Annals of Computer Science and Information Systems
Volume 25

Proceedings of the 16th Conference on Computer Science and Intelligence Systems

September 2–5, 2021. Online

Maria Ganzha, Leszek Maciaszek, Marcin Paprzycki,
Dominik Ślęzak (eds.)

Annals of Computer Science and Information Systems, Volume 25

Proceedings of the 16th Conference on Computer Science and Intelligence Systems

Maria Ganzha, Leszek Maciaszek, Marcin Paprzycki, Dominik Ślęzak (eds.)

2021, Warszawa,
Polskie Towarzystwo
Informatyczne

2021, New York City,
Institute of Electrical and
Electronics Engineers

Annals of Computer Science and Information Systems, Volume 25

Proceedings of the 16th Conference on Computer Science and Intelligence Systems

ART: ISBN 978-83-959183-8-4, IEEE Catalog Number CFP2185N-ART
USB: ISBN 978-83-959183-7-7, IEEE Catalog Number CFP2185N-USB
WEB: ISBN 978-83-959183-6-0

ISSN 2300-5963
DOI 10.15439/978-83-959183-6-0

Ul. Solec 38/103
00-394 Warsaw, Poland

Contact: secretariat@fedcsis.org
http://annals-csis.org/

Cover art: Elbląg Zdrój
Adriana Rонżewska-Kotyńska,
Elbląg, Poland

Also in this series:

Volume 26: Position and Communication Papers of the 16th Conference on Computer Science and Intelligence Systems, **ISBN WEB: 978-83-959183-9-1, ISBN USB: 978-83-962423-0-3**
Volume 24: Proceedings of the International Conference on Research in Management & Technovation 2020, **ISBN WEB: 978-83-959183-5-3, ISBN USB: 978-83-959183-4-6**
Volume 23: Communication Papers of the 2020 Federated Conference on Computer Science and Information Systems, **ISBN WEB: 978-83-959183-2-2, ISBN USB: 978-83-959183-3-9**
Volume 22: Position Papers of the 2020 Federated Conference on Computer Science and Information Systems, **ISBN WEB: 978-83-959183-0-8, ISBN USB: 978-83-959183-1-5**
Volume 21: Proceedings of the 2020 Federated Conference on Computer Science and Information Systems, **ISBN Web 978-83-955416-7-4, ISBN USB 978-83-955416-8-1, ISBN ART 978-83-955416-9-8**
Volume 20: Communication Papers of the 2019 Federated Conference on Computer Science and Information Systems, **ISBN WEB: 978-83-955416-3-6, ISBN USB: 978-83-955416-4-3**
Volume 19: Position Papers of the 2019 Federated Conference on Computer Science and Information Systems, **ISBN WEB: 978-83-955416-1-2, ISBN USB: 978-83-955416-2-9**
Volume 18: Proceedings of the 2019 Federated Conference on Computer Science and Information Systems, **ISBN Web 978-83-952357-8-8, ISBN USB 978-83-952357-9-5, ISBN ART 978-83-955416-0-5**
Volume 17: Communication Papers of the 2018 Federated Conference on Computer Science and Information Systems, **ISBN WEB: 978-83-952357-0-2, ISBN USB: 978-83-952357-1-9**
Volume 16: Position Papers of the 2018 Federated Conference on Computer Science and Information Systems, **ISBN WEB: 978-83-949419-8-7, ISBN USB: 978-83-949419-9-4**
Volume 15: Proceedings of the 2018 Federated Conference on Computer Science and Information Systems, **ISBN WEB: 978-83-949419-5-6, ISBN USB: 978-83-949419-6-3, ISBN ART: 978-83-949419-7-0**
Volume 14: Proceedings of the First International Conference on Information Technology and Knowledge Management, **ISBN WEB: 978-83-949419-2-5, ISBN USB: 978-83-949419-1-8, ISBN ART: 978-83-949419-0-1**

DEAR Reader, it is our pleasure to present to you Proceedings of the 16th Conference on Computer Science and Intelligence Systems (FedCSIS'2021), which took place, fully remotely, on September 2-4, 2021. Conference was originally planned to take place in Sofia, Bulgaria, but the global COVID-19 pandemics, again, forced us to adapt and organize the conference online.

Before proceeding further, let us share a very important information. In June 2021 FedCSIS conference series has been ranked B in the CORE ranking system. This constitutes a major achievement for the series. This is particularly valuable achievement since the series was not ranked before. We would like to thank prof. Paweł Sitek for leading our efforts and preparing all necessary documentation.

FedCSIS'2021 was chaired by prof. Stefka Fidanova, while dr. Nina Dobrinkova acted as the Chair of the Organizing Committee. This year, FedCSIS was organized by the Polish Information Processing Society (Mazovia Chapter), IEEE Poland Section Computer Society Chapter, Systems Research Institute Polish Academy of Sciences, Warsaw University of Technology, Wrocław University of Economics and Business, and Institute of Information and Communication Technologies, Bulgarian Academy of Sciences.

FedCSIS'2021 was technically co-sponsored by: IEEE Bulgarian Section, IEEE Poland Section, IEEE Czechoslovakia Section Computer Society Chapter, IEEE Poland Section Systems, Man, and Cybernetics Society Chapter, IEEE Poland Section Computational Intelligence Society Chapter, IEEE Poland Section Control System Society Chapter, Committee of Computer Science of the Polish Academy of Sciences, Mazovia Cluster ICT, Poland, Eastern Cluster ICT, Poland and Bulgarian Section of SIAM.

We also glad to announce that this year (and we believe that in future also) the FedCSIS conference formed strategic alliance with QED Software, a Polish software company developing AI-based products.

During FedCSIS'2021, the keynote lectures were delivered by:

- David Bader, Distinguished Professor, New Jersey Institute of Technology: "*Solving Global Grand Challenges with High Performance Data Analytics"*
- Rajkumar Buyya, Director, Cloud Computing and Distributed Systems (CLOUDS) Lab, The University of Melbourne, Australia and CEO, Manjrasoft Pvt Ltd, Melbourne, Australia: "*Neoteric Frontiers in Cloud and Edge Computing*"
- Hristo Djidjev, Los Alamos National Laboratory: "*Using quantum annealing for discrete optimization"*
- Moshe Y Vardi, Professor, Rice University: *"Lessons from COVID-19: Efficiency vs Resilience"*

FedCSIS 2021 consisted of five Tracks and one special event for Young Researchers. Within each Track, topical Technical Sessions have been organized. Some of these Technical Sessions have been associated with the FedCSIS conference series for many years, while some of them are relatively new. Their role is to focus and enrich discussions on selected areas pertinent to the general scope of each Track. Here is the list of tracks and topical Technical Sessions organized within their scope.

- **Track 1 Artificial Intelligence in Applications (16th Symposium AAIA'21)**
  - Computational Optimization (14th International Workshop WCO'21)
- **Track 2: Computer Science & Systems (CSS'21)**
  - Computer Aspects of Numerical Algorithms (14th Workshop CANA'21)
  - Multimedia Applications and Processing (14th International Symposium MMAP'21)
- **Track 3: Network Systems and Applications (NSA'21)**
  - Internet of Things – Enablers, Challenges and Applications (5th Workshop IoT-ECAW'21)
  - Cyber Security, Privacy, and Trust (2nd International Forum NEMESIS'21)
- **Track 4: Advances in Information Systems and Technology (AIST'21)**
  - Data Science in Health, Ecology and Commerce (3rd Special Session DSH'21)
  - Information Systems Management (16th Conference ISM'21)
  - Knowledge Acquisition and Management (27th Conference KAM'21)
- **Track 5: Software and System Engineering (S3E'21)**
  - Cyber-Physical Systems (8th International Workshop IWCPS-8)
  - Software Engineering Workshop (41st IEEE Workshop SEW-41)
- **Artificial Intelligence and Cybersecurity (1st Young Researchers Workshop YRW'21)**

Each paper, found in this volume, was refereed by at least two referees and the acceptance rate of regular full papers was ~24.8% (32 regular full papers out of 129 general submissions).

During FedCSIS 2021, for the first time, the Professor Zdzisław Pawlak award was elevated from the AIA Track award to the award presented to the best papers across the whole conference. It was done to further integrate the conference, following the fact that scientific achievements of Professor Pawlak had gone far beyond artificial intelligence. This year the following awards have been given:

- In the category **Best paper** – Anh Nguyen Mac, Hung Son Nguyen for the paper entitled "*Rotation Variance in Graph Convolutional Networks*"
- In the category **Young Researcher** – Christian Leyh, Konstanze Köppel, Sarah Neuschl, Milan Pentrack for the paper entitled "*Critical Success Factors for Digitalization Projects*"
- In the category **Industry cooperation** – Lov Kumar, Mukesh Kumar, Lalita Bhanu Murthy, Sanjay Misra, Vipul Kocher, Srinivas Padmanabhuni for the paper entitled "*An Empirical Study on Application of Word Embedding*

Techniques for Prediction of Software Defect Severity Level"

- In the category **International Cooperation** – Arman Ferdowsi, Alireza Khanteymoori for the paper entitled "*Discovering Communities in Networks: A Linear Programming Approach Using Max-Min Modularit*y"

The program of FedCSIS required a dedicated effort of many people. We would like to express our warmest gratitude to all Committee members, of each Track and each Technical Session, for their hard work in attracting and later refereeing 129 submissions.

We thank the authors of papers for their great contribution into the theory and practice of computing and software systems. We are grateful to the invited speakers for sharing their knowledge and wisdom with the participants.

We hope that you had an inspiring conference. We also hope to meet you again for the 17th Conference on Computer Science and Intelligence Systems (FedCSIS 2022). This time we are almost certain that we will be able to organize the conference on site and will finally reach Sofia, Bulgaria.

Co-Chairs of the FedCSIS Conference Series:

Maria Ganzha, *Warsaw University of Technology, Poland and Systems Research Institute Polish Academy of Sciences, Warsaw, Poland*

Leszek Maciaszek, *Wrocław University of Economics and Business, Wrocław, Poland and Macquarie University, Sydney, Australia*

Marcin Paprzycki, *Systems Research Institute Polish Academy of Sciences, Warsaw Poland and Management Academy, Warsaw, Poland*

Dominik Ślęzak, *Institute of Informatics, University of Warsaw, Poland*

Annals of Computer Science and Information Systems,
Volume 25

Proceedings of the 16th Conference on Computer Science and Intelligence Systems

September 2–5, 2021. Online

TABLE OF CONTENTS

INVITED CONTRIBUTIONS

15TH INTERNATIONAL SYMPOSIUM ADVANCES IN ARTIFICIAL INTELLIGENCE AND APPLICATIONS

SHORT PAPERS

14TH INTERNATIONAL WORKSHOP ON COMPUTATIONAL OPTIMIZATION

SHORT PAPERS

COMPUTER SCIENCE AND SYSTEMS

14TH WORKSHOP ON COMPUTER ASPECTS OF NUMERICAL ALGORITHMS

SHORT PAPERS

14TH INTERNATIONAL SYMPOSIUM ON MULTIMEDIA APPLICATIONS AND PROCESSING

NETWORK SYSTEMS AND APPLICATIONS

5th Workshop on Internet of Things - Enablers, Challenges and Applications

2nd International Forum on Cyber Security, Privacy and Trust

Advances in Information Systems and Technology

3rd Special Session on Data Science in Health, Ecology and Commerce

16th Conference on Information Systems Management

SHORT PAPERS

27TH CONFERENCE ON KNOWLEDGE ACQUISITION AND MANAGEMENT

SHORT PAPERS

SOFTWARE, SYSTEM AND SERVICE ENGINEERING

SHORT PAPERS

JOINT 41ST IEEE SOFTWARE ENGINEERING WORKSHOP AND 8TH INTERNATIONAL WORKSHOP ON CYBER-PHYSICAL SYSTEMS

SHORT PAPERS

DOI: 10.15439/2021F001
ISSN 2300-5963 ACSIS, Vol. 25

An Efficient Connected Swarm Deployment via Deep Learning

Kiril Danilchenko
School of Electrical and Computer Engineering
Ben-Gurion University of the Negev
Beer-Sheva, Israel
kirild@post.bgu.ac.il

Michael Segal
School of Electrical and Computer Engineering
Ben-Gurion University of the Negev
Beer-Sheva, Israel
segal@bgu.ac.il

***Abstract*—In this paper, an unmanned aerial vehicles (UAVs) deployment framework based on machine learning is studied. It aims to maximize the sum of the weights of the ground users covered by UAVs while UAVs forming a connected communication graph. We focus on the case where the number of UAVs is not necessarily enough to cover all ground users.**

We develop an UAV Deployment Deep Neural network (*UDDNNet*) as a UAV's deployment deep network method. Simulation results demonstrate that *UDDNNet* can serve as a computationally inexpensive replacement for traditionally expensive optimization algorithms in real-time tasks and outperform the state-of-the-art traditional algorithms.

I. Introduction

IN THIS work we consider the case where Unmanned Aerial Vehicles (UAVs) provide coverage to ground users within the serving area. Thus, each UAV serves as an aerial Base Station (BS). Different researchers introduced different objectives and proposed different aerial BSs deployment algorithms in order to solve them. The objective function of these algorithms varied from maximizing the system capacity to minimizing the required number of aerial BSs.

In the last years, the researchers have focused on deployment optimization algorithms that minimize the required number of aerial BSs providing wireless coverage to all ground users. Opposite to most existing work, where the number of UAVs is assumed to be enough to cover all ground users [13], [14], [17], in this study, we consider a more realistic scenario: UAVs are given, and their number is not necessarily enough to cover all ground users. This assumption is reasonable in any practical scenario. For example, in emergency cases, the number of available UAVs is limited, but the number of first responders (firefighters, medicals, etc.) is larger than the UAVs can cover.

In this work, we assume that each ground user has a *rank* that defines the importance of the user's coverage. The rank and location of ground users can be changed from time to time. Thus, we are facing with the question: "Who should be covered and who should not, at any point in time?" For a given particular time snapshot, this problem is known to be NP-hard [4]. Additionally, we require the local connectivity between the UAVs among themselves (see Figure 1) and not through some global entity that connects between them.

We adopt the model and problems first proposed in [3]. Formally, we consider a set S of n points distributed in the plane, where each point $s_i \in S, i = 1, \dots, n,$ has a positive weight $w(s_i)$. We assume that all UAVs fly at the same fixed altitude. All UAVs fly at the same fixed altitude, and covering disks on the ground have the same radius. Denote this radius as R_{COV}, and each UAV has a communication radius (R_{COM}). Now, we define the covering problems formally where UAVs provide connectivity between themselves. Consider a set P of m disks (represent the covering disks of the UAV's) of radius R_{COV}, where set C contains the centers of these disks.

Connected max(S,m)($Cmax(S, m)$): Given a set S and a parameter R_{COM}, place the disks from the set P such that:

1) The total weight of points from the set S covered by the disks is maximized.
2) The undirected graph $G = (C, E)$ imposed on P should be connected, where an edge $(u, v) \in E$ if $d(u, v) \leq R_{COM}$, for $d(u, v)$ being the L_2 distance between the centers $v, u \in C$.

Similarly to the above, the **Connected-Dynamic max(S,m)** ($CDmax(S, m)$) problem aims to maintain the disks from P under dynamic updates of S.

State-of-the-art solutions often involve exhaustive searches or the optimization of various heuristics. We tackle our problem from a different perspective; we leverage recent deep learning (DL) advances to design a novel deep neural network (DNN) architecture to provide better performance with reduced runtime. The proposed DNN approach establishes a connection between the total weight cover maximization problems under connectivity constraints while minimizing a loss function during DNN training. Additionally, it relies on an efficient network training and ensemble mechanism to beat state-of-the-art solutions.

The main contributions of this work can be summarized as follows. First, we propose a UAV's deployment strategy using a DNN. To this end, we offer a novel DNN structure trained on the optimal solutions to a target problem. We use a supervised learning approach to solve our problem compared to another approaches that use the unsupervised or reinforcement methods. The uniqueness of our study lies in the fact that our training set is constructed being based on optimal solutions for this problem. Therefore, the solutions derived by our DNN

IEEE Catalog Number: CFP2185N-ART ©2021, PTI

have efficiency close to that of an optimal solution. Second, the performance of the proposed DNN approach is verified through extensive evaluation. The simulation results confirm that the proposed DNN achieves outstanding approximation solutions to the target problem with shorter than the state-of-the-art evaluated solutions computation times.

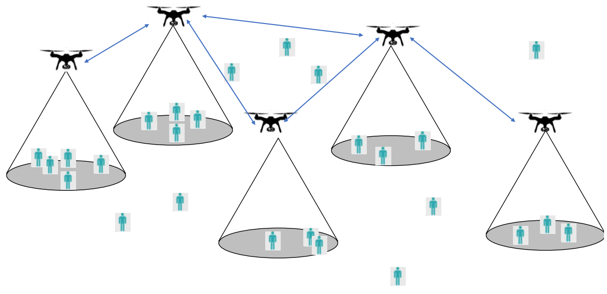


Fig. 1: The UAVs provide connectivity among themselves.

The remainder of this paper is organized as follows. Next Section II discusses recent related studies. The details of the proposed Deep Learning architecture are described in Section III.In section IV we summarized all notation used in this paper. In Section V we describe all aspects of evaluation setups. The simulation results are described in Section VI. Finally, conclusions and suggestions for future research are presented in Section VII.

II. Related Work

In this section, we first briefly review the typical approaches. There is a growing number of researches on the topic of UAV-based stations (aerial-BS) placement. UAV deployment under different constrains have been widely discussed in recent years and resulted in the development of various heuristics [13], [14], [17], [22], [23], [20], [20]. The problem of placing m disks that cover a maximal weight of given points (without connectivity requirement) has been given some attention in the past. The authors of [4] presented the problem of covering the maximum number of points in the point set S with m unit disks, without the demand for disks connectivity. They gave a $(1-\varepsilon)$-approximation algorithm with time complexity $O(n\varepsilon^{-4m+4})\log^{2m-1}(\frac{1}{\varepsilon})$. The problem to place m rectangles such that the sum of the weights of the points in S covered by these rectangles is maximized is considered in [10]. For any fixed $\varepsilon > 0$, the authors present efficient approximation schemes that can find a $(1-\varepsilon)$-approximation to the optimal solution in $O(\frac{n}{\varepsilon}\log(\frac{1}{\varepsilon})+m(\frac{1}{\varepsilon})^{O(\min(\sqrt{m},\frac{1}{\varepsilon}))})$ runtime. In [6] the authors presented a PTAS for a more general case different covering shapes (disks, polygons with $O(1)$ edges), running in $O(n\frac{1}{\varepsilon}^{O(1)}+\frac{m}{\varepsilon}\log m+m(\frac{1}{\varepsilon})^{O(\min(m,\frac{1}{\varepsilon}))})$ time. The authors of [19] solve the relevant problem to place two disks in the plane to ensure both maximal covering and full connectivity by providing two algorithms having $O(n^4)$ and $O(n^3\log n)$ time complexity, respectively. Another related problem is presented in [9], [5]. In these works, the authors formulate the following problem: given a set of n discs in the plane, select a subset of k disks that maximize the area of their union, under the constrain that this union is connected. The authors of [2] consider the $Cmax(S,m)$. They gave $O(\frac{1}{\sqrt{m}})$ with time complexity $O(\beta^2 mn\log n)$, where $\beta\cdot R_{COM}=d_{max}$ and d_{max} be the largest L_∞ distance defined by a pair of points in S. In [3] the authors gave $O(1)$ approximation for $Cmax(S,m)$, and presented an algorithm for $CDmax(S,m)$ using $O(m\sqrt{m})$ UAVs with the approximation ratio $O(1)$.

We continue with researches that use Machine Learning techniques. The authors of [8] aim how to maximize the number of users covered by the system in an emergency scenario. They proposed the use of RL (Q-learning) to determine the optimal position of the UAVs. The proposed solution was compared to different positioning strategies and outperformed all other methods in all considered metrics. In [16] the authors considered the problem of the optimal deployment of multiple UAVs to maximize throughput for ground users with different requirements. The authors use Reinforcement Learning (RL) to calculate the locations of the UAVs. Qiu et al. [18] considered the problem of maximizing the coverage rate of N ground users by the simultaneous placement of multiple UAVs with a limited coverage range. They applied the Deep Reinforcement Learning method to cope with this problem. Liu et al. [11] proposed a deep RL (DRL), a method for energy-efficient UAV control to provide communication coverage for ground users. The control policy considers the UAV movements in each time slot, and the aim is to optimize the communication coverage, fairness, energy consumption, and connectivity. Liu et al. [12] developed a fast positioning algorithm for the deployment of aerial BSs, where the objective is to maximize the sum of the downlink rates in the multiple UAV communication network. They designed a geographical position information (GPI) learning algorithm. Dai et al. [1] investigated the problem of the efficient deployment of UAVs in order guarantee the quality-of-service requirements. The UAV played the role of a coordinator to provide high-quality communication service for ground users and maximize the benefits of caching. The authors proposed an RL-based approach to solving the multi-objective deployment problem while maintaining an optimal tradeoff between the power consumption and backhaul saving. They adopted the RL approach to determine the 3D placement and minimum transmit power, and cache strategy of each UAV.

In summary, recent studies have used DL to solve aerial BS (UAVs) deployment under different objectives. We propose a novel method of using DL to solve the UAVs deployment such that the UAVs cover a maximal total weight of ground users under the connectivity requirement between UAVs. We use a supervised approach to solve our problem compared to other researches using the unsupervised or reinforcement methods. The uniqueness of our study lies in the fact that our training set is constructed basing on optimal solutions for this problem. Therefore, the solutions derived by our DNN have efficiency close to that of an optimal solution.

III. *UDDNNet* STRUCTURE

In this section, we describe the proposed *UDDNNet*, including the details of the DNN design and a training process based on supervised learning.

In the following, we detail the proposed DNN architecture and discuss how training and testing are performed.

1) Network Structure: Our proposed approach uses a fully connected neural network with two input layers, $L = 8$ fully connected hidden layers, and one output layer.

The first input for the proposed network is a matrix with dimensions of $3 \times n$, where entry i of the matrix represents a location and a weight of ground user i.

The second input is a binary matrix with dimensions of $2 \times \mathcal{K}$, where a number of rows without zero elements equals the maximal number of UAVs that is possible to use in this scenario. Note that the $\mathcal{K}$ represents the maximum value of m in our experiments. Denote this matrix as *FilMat*. We use this input as a binary filter matrix, meaning a non-zero entry in the matrix signifies that we can put an UAV in this location. Therefore, we use this matrix to avoid a scenario in which the DNN located more UAVs than given in advance.

The first hidden layer reshapes the input matrix into a one-dimensional vector with a length of 5000. The second hidden layer reshapes its input into a one-dimensional vector with a length of 4000. In this manner, the following five hidden layers perform reshaping until the output has dimensions of 100. The 8^{-th} hidden layer reshapes the one-dimensional vector with a length of 100 into a matrix with dimensions of $2 \times \mathcal{K}$. This matrix multiplied by *FilMat* and the result of this multiplication is an input of the output layer, where a activation function Eq. 1 was applied. The output of the network is the location of m UAV's.

The $ReLU(x)$ function is used as the activation function for the hidden layer, where $ReLU(x)$ is the rectified linear unit function $\max(x, 0)$ [15]. Additionally, to enforce the location constraint we adopt a special activation function from [21] for the output layer of the DNN, as shown below.

$$y(x) = \min(ReLU(x), \sqrt{A}), \qquad (1)$$

We apply this activation function in the output layer to limit the output location to the range of $[0, \sqrt{A}]$, where A is the square zone of interest area.

We let l_k to denote the number of neurons in the k^{-th} layer. The k^{-th} layer is a hidden layer and its output is calculated as follows:

$$c_k = ReLU(W_k \cdot c_{k-1}), \qquad (2)$$

where c_{k-1} and c_k are the output vectors of the previous and current layers with dimensions of $l_{k-1} \times 1$ and $l_k \times 1$, respectively. W_k is the $l_k \times l_{k-1}$ weight matrix.

A detailed explanation of the DNN architecture is provided in Fig. 2. The motivation of the proposed DNN architecture is to "shrink" the inputs (location and weight of the ground users) into m two dimensional coordinates, the locations of the UAVs.

IV. NOTATIONS

The notations used in this paper are summarized in Table I

Symbol	Meaning
S	The set of n ground users
C	The set of the centers of the disks
R_{COV}	The covering disk radius in the case of
R_{COM}	The communication radius
m	The number of available UAVs
A	The area of zone of interest
$\mathcal{K}$	The maximum value of m

TABLE I: Summary of notations used in this study.

V. EXPERIMENTAL SETUP

To evaluate the proposed *UDDNNet* performance, we conducted experiments with a different number of ground users and the different number of available UAVs. This section describes the data generation process, splitting of whole data set to training, validation and testing sets, training details, and testing process.

1) Data Generation: The *UDDNNet* was trained using optimal solutions implemented in the Wolfram Language. Data was generated in the following manner.

First, we randomly distribute m disks in an area of interest with dimensions of 5000×5000 m^2, such that the disk graph imposed on their centers is connected. We set m to be $m \sim U[2, \ldots, \mathcal{K}]$, where $\mathcal{K} = 10$. Next, we distributed on these disks between $10\% - 30\%$ of ground users. Finally, we randomly distributed the ground users in an area of interest. Also, for each ground user we randomly assign a weight $w(s_i) \sim U[0, 1]$.

We repeated the process described above multiple times to generate a dataset. The final dataset contained approximately 100000 instances. We randomly split the dataset into three sets for training, validation and testing, where the sizes of each set were 70%, 10% and 20% of the entire dataset, respectively.

2) Training Process: We used the entire *training dataset* to optimize the weights of the neural network. The loss function we adopted was the mean absolute error (MAE) as a loss between the optimal UAV's location and the network's output. We used the ADAM optimizer [7] for optimization. We analyzed the impact of the batch size and learning rate of *UDDNNet*. Based on the results presented in Fig.3 and Fig.4, we selected a batch size of 256 and the learning rate was set to 0.0001.

In Fig. 5 we can see the the training error and the validation error as a function of the training epoch (rounds) with the parameters chosen in Fig.3 and Fig.4. We can see that a validation error decreases when the number of rounds is increased.

3) Testing Process: In the testing stage, we used the testing dataset, passed each instance through the trained *UDDNNet*, and collected the results-location of the UAVs. We then compared the resulting total covering weight by UAVs achieved

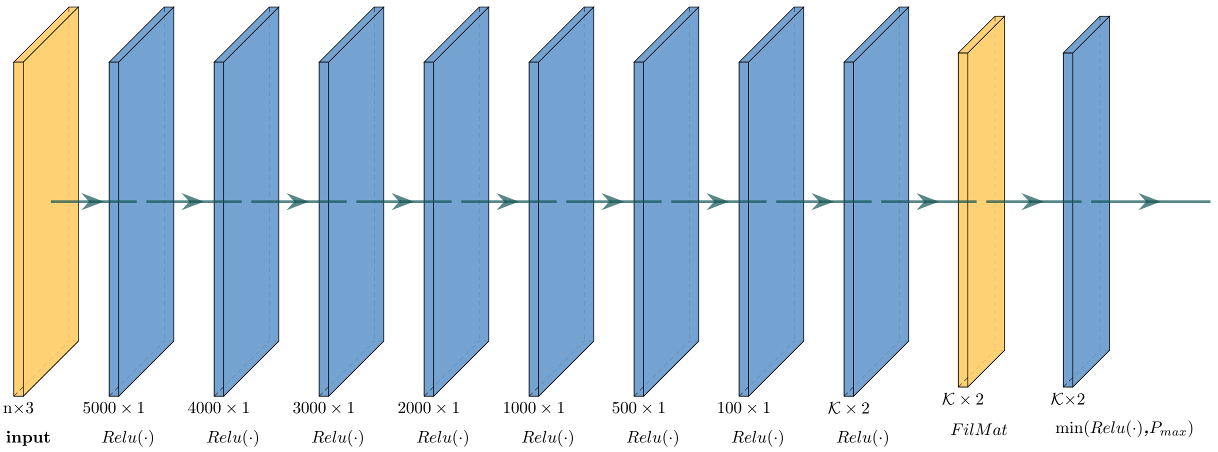


Fig. 2: DNN architecture for UAVs deployment.

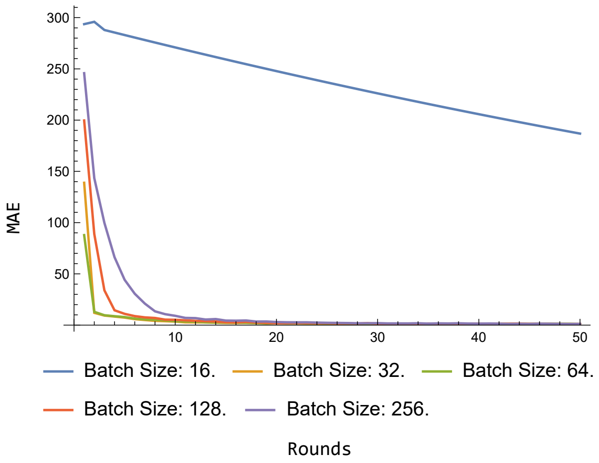


Fig. 3: Batch size selection

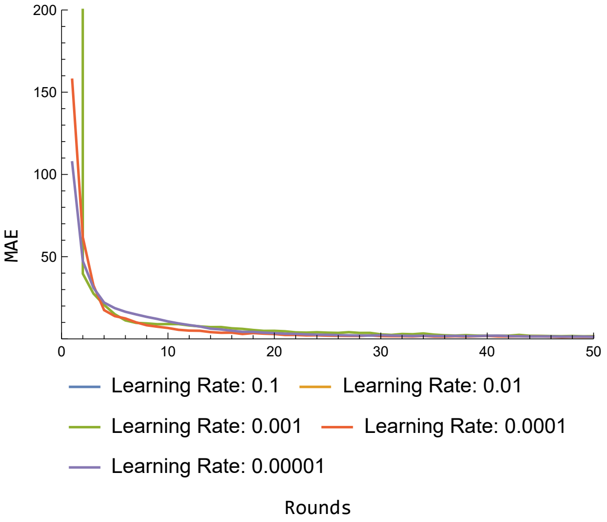


Fig. 4: Learning rate selection

by the compression scheme and the solution based on the locations generated by *UDDNNet*.

4) Schemes for Comparison: Besides *UDDNNet* we also implemented the algorithms presented in [3]. The authors divide the area of interest into a grid with cell size r. They represent each cell as a node in the graph with a weight equal to the total sum of ground users' weights belong to this cell. They gave $O(1)$ approximation for $Cmax(S, m)$and presented an algorithm for $CDmax(S, m)$ using $O(m\sqrt{m})$ UAVs to keep the approximation ratio $O(1)$, where each update takes $O(\log n)$ runtime. We compare the performance of *UDDNNet* with the algorithms that solve $Cmax(S, m)$ and $CDmax(S, m)$ from [3].

VI. Evaluation Results

The proposed DNN approach was implemented in Wolfram 12.3 on a single desktop computer with the hardware specifications listed below.

1) Intel CPU Core i7-8700K @ 3.70 GHz
2) Nvidia GPU GeForce GTX 1080Ti

The GPU was used in the training stage to reduce training time but was not used in the testing stage.

A. Numerical Results

We conducted numerical simulations to verify the effectiveness of the *UDDNNet* and compare it to the heuristic presented in [3]. Detailed simulations allowed us to study the performance of the proposed *UDDNNet*, which is defined as the total weight of covered users and the runtime required by *UDDNNet*. Specifically, we examined the performance obtained by *UDDNNet* for different numbers of ground users and available UAVs to solve $CDmax(S, m)$ and $CDmax(S, m)$. Table II gathers the parameters that, unless otherwise speci-

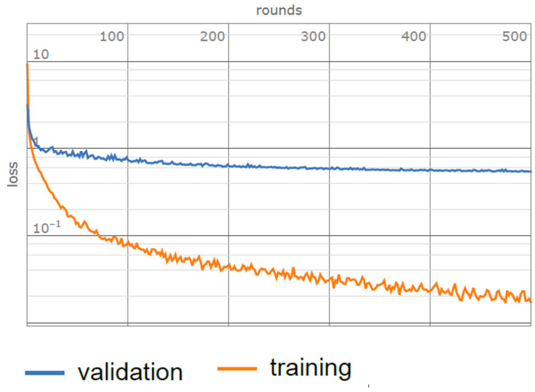


Fig. 5: Traning Procces

fied, we have used for the network model, regardless of the simulation environment.

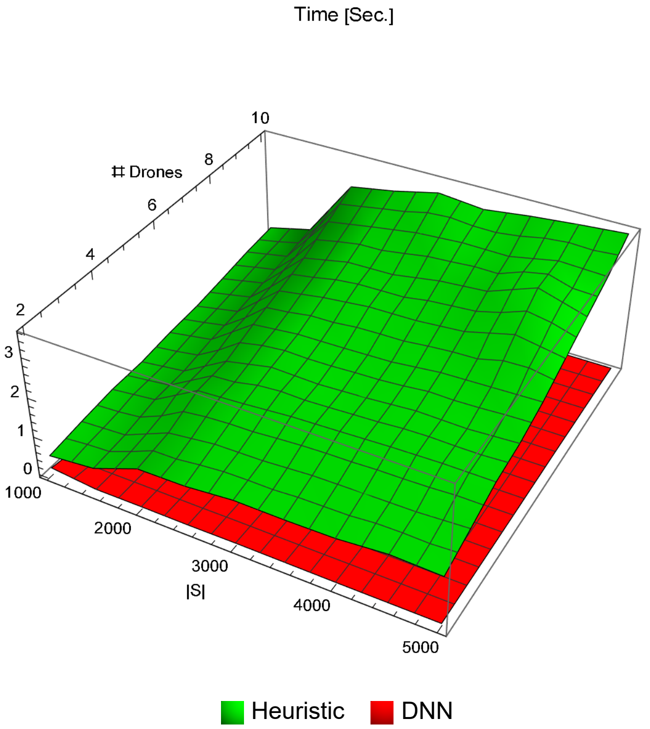


Fig. 6: Time complexity of *UDDNNet* v.s. $Cmax(S, m)$ heuristics from [3].

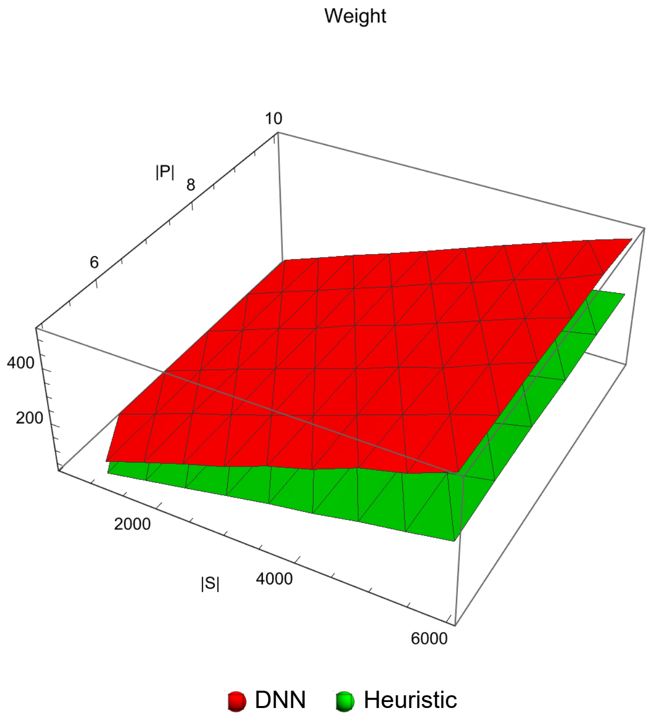


Fig. 7: Weight covered by *UDDNNet* and $Cmax(S, m)$ heuristics from [3].

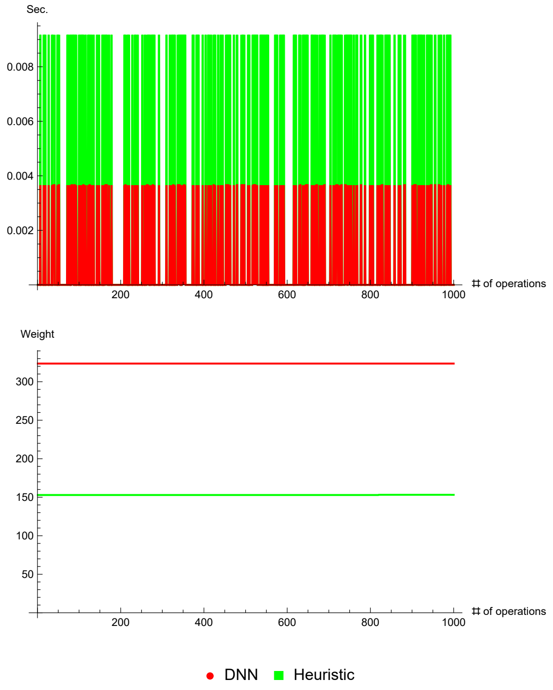


Fig. 8: Maintenance of Dynamic Covering Set. The number of ground users is 3000 and the number of UAVs is 3.

We start by examining our method's performance in the static version $Cmax(S, m)$, for the case where the node i weight is uniformly distributed $w_i = U \sim [0, 1]$. The results of this examination we can see in Fig. 7 and Fig.6 showing the superiority of *UDDNNet* approach. In particular, Figure 7 presents the total weight of covered users achieved by *UDDNNet* versus the solution from [3]. In this Figure, one can see that *UDDNNet* achieves better performance than the solution from [3] for a problem with different numbers of nodes and available UAVs. In Figure 6, we present the running time of *UDDNNet* versus that of heuristic solution from

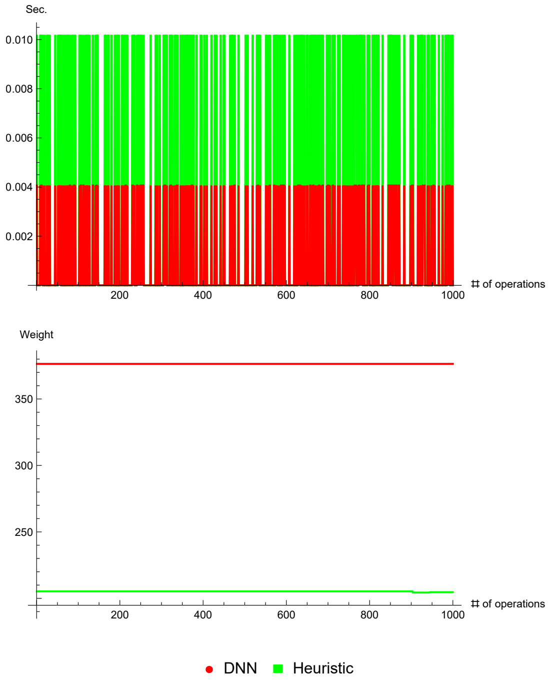


Fig. 9: Maintenance of Dynamic Covering Set. The number of ground users is 3500 and the number of UAVs is 4.

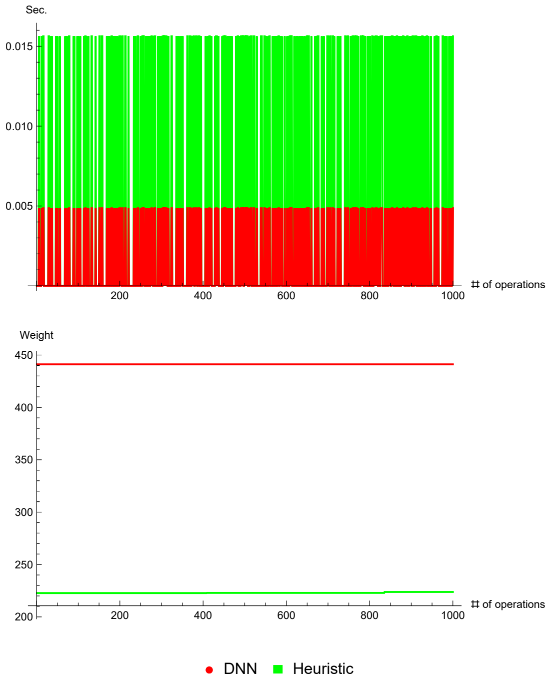


Fig. 10: Maintenance of Dynamic Covering Set. The number of ground users is 400 and the number of UAVs is 6.

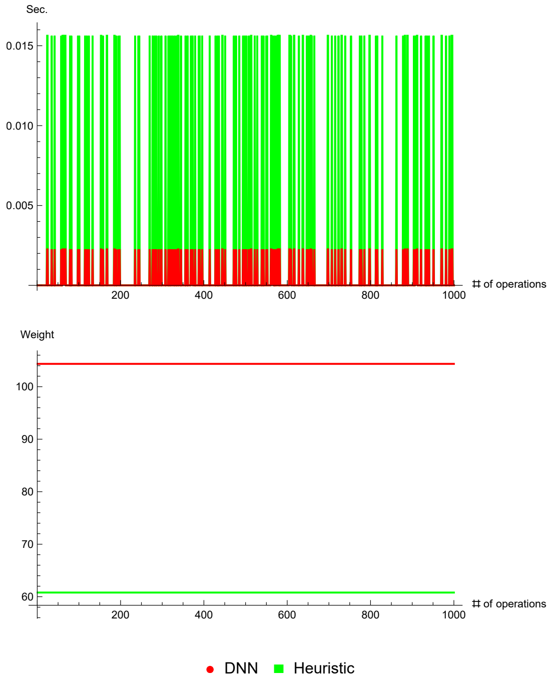


Fig. 11: Maintenance of Dynamic Covering Set. The number of ground users is 1000 and the number of UAVs is 2.

Parameter	Value
Number of ground users	$1000-5000$
Ground user's weights	$w(s_i) \sim U[0,1]$
Number of UAVs (drones)	$2-10$
Simulation Playground Size	$5000 \ \times 5000$ m^2
R_{COV}	100 m
R_{COM}	200 m
m	$2-10$

TABLE II: Simulation Configuration

[3]. One can see that *UDDNNet*'s run-time is approximately constant and is a magnitude lower than that of the solution from [3] that we use as a baseline for a problem with different numbers of nodes and available UAVs.

Now we deal with the dynamic version $CDmax(S,m)$. We examined the performance obtained by [3] and *UDDNNet* for different configurations, where we allow the use of a different number of ground users and UAVs. Note that, at each change of S, we solve the problem from scratch by *UDDNNet*. Therefore, *UDDNNet* solves at each change the static version of the given set S.

The Figures 8-11 represent the maintenance of set P under insertions or deletions of a point from set S. Each subfigure of these figures includes two graphs. The top graph represents the time complexity of dynamic maintenance of solution from [3] versus the time complexity of *UDDNNet*. The bottom graph represents the total weight covered by *UDDNNet* and out scheme for comparison. In both graphics, axis x represents the trace of 1000 operations on set S, where each operation may be insertion or deletion. Additionally, the underneath graph's axis y represents the total weight covered by *UDDNNet* and solution from [3], and the y axis in the left graph represents the time required to execute the operation. We again can witness the better performance of *UDDNNet* in terms of runtime and obtained weight of covered ground users.

VII. Conclusions and Future Work

In this study, we considered the connected version of the covering problem motivated by the coverage of ad-hoc UAVs swarm. Inspired by recent advances in artificial intelligence, we proposed the use of Deep Learning to address this problem. We developed a fully connected multi-layer neural network that takes a ground user's location and a number of available UAVs as inputs and outputs the location of the UAVs. A supervised learning strategy was adopted to train *UDDNNet* by using optimal solutions as a training dataset.

Our results are encouraging in many respects. The time complexity of the proposed DNN solutions is the most important factor among our results. Therefore, the key takeaway from our research is that a DNN can serve as a computationally inexpensive component to replace expensive optimization algorithms for real-time tasks while providing very good performance compared to state-of-the-art methods.

Acknowledgments

This research has been supported by a grant from Pazy Foundation.

References

[1] Haibo Dai, Haiyang Zhang, Baoyun Wang, and Luxi Yang. The multi-objective deployment optimization of uav-mounted cache-enabled base stations. *Physical Communication*, 34:114–120, 2019.

[2] Kiril Danilchenko and Michael Segal. Connected ad-hoc swarm of drones. In *Proceedings of the 6th ACM Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*, DroNet '20, New York, NY, USA, 2020. Association for Computing Machinery.

[3] Kiril Danilchenko, Michael Segal, and Zeev Nutov. Covering users by a connected swarm efficiently. In *International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics*, pages 32–44. Springer, 2020.

[4] Mark De Berg, Sergio Cabello, and Sariel Har-Peled. Covering many or few points with unit disks. *Theory of Computing Systems*, 45(3):446–469, 2009.

[5] Chien-Chung Huang, Mathieu Mari, Claire Mathieu, Joseph SB Mitchell, and Nabil H Mustafa. Maximizing covered area in the euclidean plane with connectivity constraint. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

[6] Kai Jin, Jian Li, Haitao Wang, Bowei Zhang, and Ningye Zhang. Near-linear time approximation schemes for geometric maximum coverage. *Theoretical Computer Science*, 725:64–78, 2018.

[7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[8] Paulo V Klaine, João PB Nadas, Richard D Souza, and Muhammad A Imran. Distributed drone base station positioning for emergency cellular networks using reinforcement learning. *Cognitive computation*, 10(5):790–804, 2018.

[9] T. Kuo, K. C. Lin, and M. Tsai. Maximizing submodular set function with connectivity constraint: Theory and application to networks. *IEEE/ACM Transactions on Networking*, 23(2):533–546, 2015.

[10] Jian Li, Haitao Wang, Bowei Zhang, and Ningye Zhang. Linear time approximation schemes for geometric maximum coverage. In *International Computing and Combinatorics Conference*, pages 559–571, 2015.

[11] Chi Harold Liu, Zheyu Chen, Jian Tang, Jie Xu, and Chengzhe Piao. Energy-efficient uav control for effective and fair communication coverage: A deep reinforcement learning approach. *IEEE Journal on Selected Areas in Communications*, 36(9):2059–2070, 2018.

[12] Jie Liu, Qiang Wang, Xuan Li, and Wenqi Zhang. A fast deployment strategy for uav enabled network based on deep learning. In *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1–6, 2020.

[13] Jiangbin Lyu, Yong Zeng, Rui Zhang, and Teng Joon Lim. Placement optimization of uav-mounted mobile base stations. *IEEE Communications Letters*, 21(3):604–607, 2016.

[14] Mohammad Mozaffari, Walid Saad, Mehdi Bennis, and Mérouane Debbah. Efficient deployment of multiple unmanned aerial vehicles for optimal wireless coverage. *IEEE Comm. Lett.*, 20(8):1647–1650, 2016.

[15] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Icml*, 2010.

[16] Yu Min Park, Minkyung Lee, and Choong Seon Hong. Multi-uavs collaboration system based on machine learning for throughput maximization. In *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pages 1–4. IEEE, 2019.

[17] Luigi Di Puglia Pugliese, Francesca Guerriero, Dimitrios Zorbas, and Tahiry Razafindralambo. Modelling the mobile target covering problem using flying drones. *Optimization Letters*, 10(5):1021–1052, 2016.

[18] Jin Qiu, Jiangbin Lyu, and Liqun Fu. Placement optimization of aerial base stations with deep reinforcement learning. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2020.

[19] Sanaz Soltani, Mohammadreza Razzazi, and Hossein Ghasemalizadeh. The most points connected-covering problem with two disks. *Theory of Computing Systems*, 62(8):2035–2047, 2018.

[20] Anand Srinivas, Gil Zussman, and Eytan Modiano. Construction and maintenance of wireless mobile backbone networks. *IEEE/ACM Transactions on Networking*, 17(1):239–252, 2009.

[21] Haoran Sun, Xiangyi Chen, Qingjiang Shi, Mingyi Hong, Xiao Fu, and Nicholas D Sidiropoulos. Learning to optimize: Training deep neural networks for interference management. *IEEE Transactions on Signal Processing*, 66(20):5438–5453, 2018.

[22] Haijun Wang, Haitao Zhao, Weiyu Wu, Jun Xiong, Dongtang Ma, and Jibo Wei. Deployment algorithms of flying base stations: 5g and beyond with uavs. *IEEE Internet of Things Journal*, 6(6):10009–10027, 2019.

[23] Xiao Zhang and Lingjie Duan. Fast deployment of uav networks for optimal wireless coverage. *IEEE Transactions on Mobile Computing*, 18(3):588–601, 2018.

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 9–15
DOI: 10.15439/2021F002
ISSN 2300-5963 ACSIS, Vol. 25

Using Free-Choice Nets for Process Mining and Business Process Management

Wil M.P. van der Aalst
Process and Data Science (Informatik 9), RWTH Aachen University, Aachen, Germany and
Fraunhofer-Institut für Angewandte Informationstechnik (FIT), Sankt Augustin, Germany
Email: wvdaalst@pads.rwth-aachen.de

***Abstract*—Free-choice nets, a subclass of Petri nets, have been studied for decades. They are interesting because they have many desirable properties normal Petri nets do not have and can be analyzed efficiently. Although the majority of process models used in practice are inherently free-choice, most users (even modeling experts) are not aware of free-choice net theory and associated analysis techniques. This paper discusses free-choice nets in the context of process mining and business process management. For example, state-of-the-art process discovery algorithms like the inductive miner produce process models that are free-choice. Also, hand-made process models using languages like BPMN tend to be free-choice because choice and synchronization are separated in different modeling elements. Therefore, we introduce basic notions and results for this important class of process models. Moreover, we also present new results for free-choice nets particularly relevant for process mining. For example, we elaborate on home clusters and lucency as closely-related and desirable correctness notions. We also discuss the limitations of free-choice nets in process mining and business process management, and suggest research directions to extend free-choice nets with non-local dependencies.**

I. Introduction

FREE-CHOICE nets can be used to model processes that include process patterns such as sequence, choice, loop, and concurrency. Compared to general Petri nets they require choice and synchronization to be separable. This is automatically the case in languages having explicit split and join operators (also called connectors or gateways) that do not mix choice and synchronization. For example, when using *Business Process Modeling Notation* (BPMN) with only AND and XOR gateways, the behavior is automatically *free-choice*. Although BPMN allows for many advanced constructs, the most widely used BPMN constructs can be easily mapped onto free-choice nets.

In this paper, we relate recent developments in free-choice nets to *Business Process Management* (BPM) in general and *process mining* in particular. The desire to manage and improve processes is not new. The field of scientific management emerged in the 1890-ties with pioneers like Frederick Winslow Taylor (1856-1915) [31]. Taylor already systematically analyzed manually recorded data in order to uncover potential process improvements. With the availability of computers, the focus shifted to automation. In the 1970-ties there was the expectation that office would would become increasingly automated, not requiring human intervention. Pioneers like Skip Ellis [18] and Michael Zisman [34] worked on so-called *office automation systems*. The ideas lead to the development of Workflow Management (WFM) systems in the 1990-ties (see [8]). Later, BPM systems broadened the scope from automation to management. In hindsight, these approaches were not so successful. For example, as the longitudinal study in [28] shows, many workflow implementations failed. As a result, WFM/BPM technology is often considered too expensive and only feasible for highly-structured processes. At the same time, people continued to model processes using flowchart-like description languages. For example, modeling tools such as ARIS and Signavio have been used to model millions of processes all over the globe. Modeling is less costly than automation, but the effect is often limited. Due to the disconnect between reality and such hand-made models, the BPM market was shrinking until recently. However, this changed with the uptake of *process mining* [2].

Process mining dramatically changed the way we look at process models and operational processes. Even seemingly simple processes like Purchase-to-Pay (P2P) and Order-to-Cash (O2C) are often amazingly complex, and traditional hand-made process models fail to capture the true fabric of such processes. Process mining bridges the gap between between *process science* (i.e., tools and techniques to improve operational processes) and *data science* (i.e., tools and techniques to extract value from data).

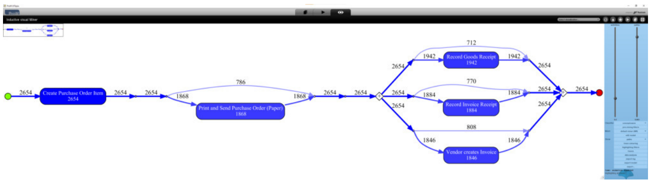


Fig. 1. Process model discovered using ProM's inductive miner.

Figure 1 shows ProM's inductive miner [22] in action. Based on (heavily filtered) data from SAP's Purchase-to-Pay (P2P) process, a process model is derived. Process discovery is just one of several process mining tasks. First, event data need to be extracted from information systems like SAP. *Process discovery* techniques transform such event data into process models (e.g., BPMN, Petri nets, and UML activity diagrams). There are simple approaches like creating so-called Directly-Follows-Graphs (DFGs) that do not discover concurrency thus having obvious problems [4]. Dozens, if not hundreds, of more sophisticated algorithms were proposed [12], [2], [13], [20], [21], [22], [33]. Using replay and alignment techniques it is possible to do *conformance checking* and relate process models (hand-made or discovered) with event data. This can be used to discover differences between reality and model [2], [16], [30]. Moreover, the model can be extended with additional perspectives, e.g., organizational aspects, decisions, and temporal aspects.

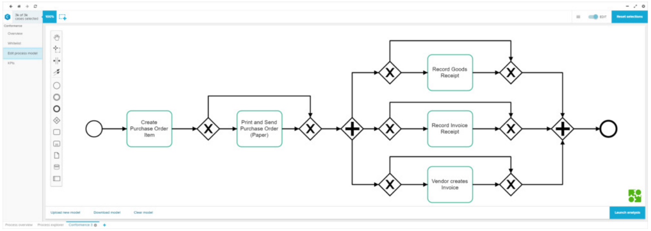


Fig. 2. BPMN model discovered using Celonis.

Currently, there are over 35 commercial process mining vendors (ABBYY Timeline, ARIS Process Mining, BusinessOptix, Celonis Process Mining, Disco/Fluxicon, Everflow, Lana, Mavim, MPM, Minit, PAFnow, QPR, etc.) and process mining is applied in most of the larger organizations. Figure 2 shows a BPMN model discovered using the Celonis process mining software. The same model can also be used for conformance checking and show where reality and model deviate.

Unlike traditional WFM/BPM technologies, there is a direct connection to the data. This allows stakeholders to spot inefficiencies, delays, and compliance problems in real-time. Process mining revitalized the BPM discipline, as is proven by the valuation of process mining firms. For example, Celonis is currently the first and only German "Decacorn" (i.e., a start-up whose value is considered to be over $10 billion).

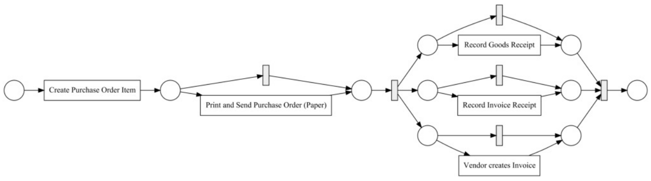


Fig. 3. A free-choice net generated from the models in Figures 1 and 2.

So how this related to free-choice nets? Process models play a key role in BPM and process mining, and these models can often be viewed as free-choice. Commonly used process notations are DFGs, BPMN models, Petri nets, and process trees. For example, the inductive mining approach uses *process trees* [22]. Although not visible, Figures 1 and 2 were actually generated using this approach. Process trees can be visualized using BPMN or Petri nets. Figure 3 shows the Petri net representation of the process tree. Any process tree corresponds to a so-called *free-choice net* having the same behavior. Later we will provide a formal definition for these notions. At this stage, it is sufficient to know that, in a free-choice net, choice and synchronization can be separated.

Any process tree can be converted to a free-choice net. Moreover, a large class of BPMN models is inherently free-choice. In a BPMN model there are flow objects such as events, activities, and gateways that are connected through directed arcs and together form a graph [26]. There are many modeling elements, but most process modelers use only a small subset [24]. For example, in many models, only exclusive gateways (for XOR-splits/joins) and parallel gateways (for AND-splits/joins) are used. Such models can be converted to free-choice nets [27]. It is also possible to convert BPMN models with inclusive gateways (i.e., OR-splits/joins) into free-choice nets (as long as the splits and joins are matching).

Since most process discovery techniques discover process models that are free-choice and also people modeling processes tend to come up with free-choice models, this is an interesting class to be studied. Therefore, this paper focuses on free-choice models. The goal is to expose people interested in BPM and process mining to free-choice-net theory.

Section II introduces preliminaries, including Petri nets, free-choice nets, and lucency. *Luceny* is a rather new notion which states that there *cannot* be two states enabling the same set of activities. Section III focuses on the class of process models having so-called *home clusters*. This class extends the class of sound models that can always terminate (e.g., no deadlocks) with the class of models that have a regeneration point. Free-choice nets with home clusters are guaranteed to be lucent. Hence, these nets are interesting for a wide range of applications and an interesting target class for process mining. Section IV discusses the limitations of free-choice nets, e.g., the inability to express non-local (i.e., long-term) dependencies. These insights may help to develop better process discovery techniques that produce more precise models. Section V concludes this paper.

II. Preliminaries

Free-choice nets are well studied [14], [15], [19], [32]. The definite book on the structure theory of free-choice nets is [17]. To keep the paper self-contained, first standard Petri net notions are introduced. If unclear, consider reading one of the standard introductions [11], [25], [29]. Most of the notations used are adopted from [6].

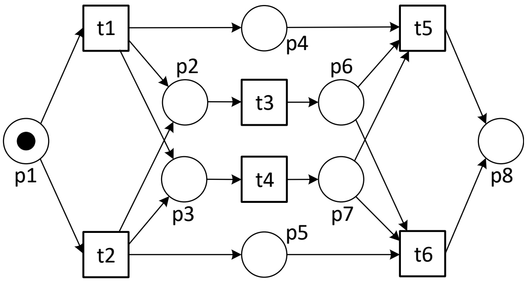


Fig. 4. A Petri net $N = (P, T, F)$ with $P = \{p1, p2, \ldots p8\}$, $T = \{t1, t2, \ldots, t6\}$, and $F = \{(p1, t1), (p1, t2), (t1, p4), \ldots, (t6, p8)\}$ that is not free-choice. The initial marking is $M = [p1]$, i.e., only place $p1$ contains a token.

A. Petri Nets

Figure 4 shows a Petri net with eight places, six transitions, and twenty arcs.

Definition 1 (Petri Net): A *Petri net* is a tuple $N = (P, T, F)$ with P the non-empty set of places, T the non-empty set of transitions such that $P \cap T = \emptyset$, and $F \subseteq (P \times T) \cup (T \times P)$ the flow relation such that the graph $(P \cup T, F)$ is (weakly) connected.

Definition 2 (Pre- and Post-Set): Let $N = (P, T, F)$ be a Petri net. For any $x \in P \cup T$: $\bullet x = \{y \mid (y, x) \in F\}$ and $x\bullet = \{y \mid (x, y) \in F\}$.

For example, in Figure 4, $\bullet p2 = \{t1, t2\}$, $\bullet t5 = \{p4, p6, p7\}$, $t1\bullet = \{p2, p3, p4\}$, and $p8\bullet = \emptyset$.

Definition 3 (Marking): Let $N = (P, T, F)$ be a Petri net. A *marking* M is a multiset of places, i.e., $M \in \mathcal{B}(P)$.[1] (N, M) is a marked net.

In the marking shown in Figure 4, transitions $t1$ and $t2$ are *enabled*. An enabled transition t can fire consuming a token from each input place in $\bullet t$ and producing a token for each output place in $t\bullet$.

Definition 4 (Enabling, Firing Rule, Reachability): Let (N, M) be a marked net with $N = (P, T, F)$. Transition $t \in T$ is enabled if $\bullet t \subseteq M$.[2] This is denoted by $(N, M)[t\rangle$ (each of t's input places $\bullet t$ contains at least one token). $en(N, M) = \{t \in T \mid (N, M)[t\rangle\}$ is the set of enabled transitions. Firing an enabled transition t results in marking $M' = (M \setminus \bullet t) \cup t\bullet$. $(N, M)[t\rangle(N, M')$ denotes that t is enabled in M and firing t results in marking M'. A marking M' is *reachable* from M if there exists a *firing sequence* σ such that $(N, M)[\sigma\rangle(N, M')$. $R(N, M) = \{M' \in \mathcal{B}(P) \mid \exists_{\sigma \in T^*}\ (N, M)[\sigma\rangle(N, M')\}$ is the set of all reachable markings. $(N, M)[\sigma\rangle$ denotes that the sequence σ is enabled when starting in marking M (without specifying the resulting marking).

Let N be the Petri net shown in Figure 4. $(N, [p1])[\sigma_1\rangle(N, [p4, p6, p7])$ with $\sigma_1 = \langle t1, t3, t4\rangle$ and $(N, [p1])[\sigma_2\rangle(N, [p8])$ with $\sigma_2 = \langle t2, t4, t3, t6\rangle$. We also define the usual properties for Petri nets.

Definition 5 (Live, Bounded, Safe, Dead, Deadlock-free, Well-Formed): A marked net (N, M) is *live* if for every reachable marking $M' \in R(N, M)$ and for every transition $t \in T$ there exists a marking $M'' \in R(N, M')$ that enables t. A marked net (N, M) is k-bounded if for every reachable marking $M' \in R(N, M)$ and every $p \in P$: $M'(p) \leq k$. A marked net (N, M) is *bounded* if there exists a k such that (N, M) is k-bounded. A 1-bounded marked net is called *safe*. A place $p \in P$ is *dead* in (N, M) when it can never be marked (no reachable marking marks p). A transition $t \in T$ is *dead* in (N, M) when it can never be enabled (no reachable marking enables t). A marked net (N, M) is *deadlock-free* if each reachable marking enables at least one transition. A Petri net N is *structurally bounded* if (N, M) is bounded for any marking M. A Petri net N is *structurally live* if there exists a marking M such that (N, M) is live. A Petri net N is *well-formed* if there exists a marking M such that (N, M) is live and bounded.

Definition 6 (Proper Petri Net): A Petri net $N = (P, T, F)$ is *proper* if all transitions have input and output places, i.e., for all $t \in T$: $\bullet t \neq \emptyset$ and $t\bullet \neq \emptyset$.

Definition 7 (Strongly Connected): A Petri net $N = (P, T, F)$ is *strongly connected* if there is a directed path between any pair of nodes.

Note that a strongly connected net is also proper. Figure 4 shows that the converse does not hold, the net is proper, but not strongly connected.

Definition 8 (Home Marking): Let (N, M) be a marked net. A marking M_H is a *home marking* if for every reachable marking $M' \in R(N, M)$: $M_H \in R(N, M')$.

The marked Petri net in Figure 4 has one home marking: $[p8]$.

B. Free-Choice Nets

The concepts and notations discussed apply to any Petri net. Now we focus on the class of *free-choice nets*. As indicated in the introduction, this is an important class because most process models used in the context of BPM and process mining are free-choice.

Definition 9 (Free-choice Net): Let $N = (P, T, F)$ be a Petri net. N is *free-choice net* if for any $t_1, t_2 \in T$: $\bullet t_1 = \bullet t_2$ or $\bullet t_1 \cap \bullet t_2 = \emptyset$.

The Petri net in Figure 4 is not free-choice because $\bullet t_5 \cap \bullet t_6 = \{p6, p7\} \neq \emptyset$, but $\bullet t_5 \neq \bullet t_6$. If we remove the places $p4$ and $p5$, then the net becomes free-choice. The places model a so-called *long-term (or non-local) dependency*: The choice between $t1$ and $t2$ in the beginning is controlling the choice between $t5$ and $t6$ at the end.

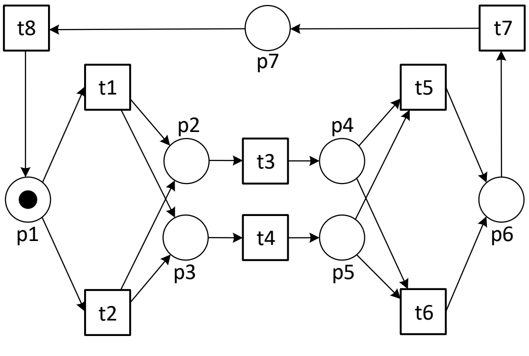


Fig. 5. A strongly-connected free-choice net.

Figure 5 is free-choice. Transitions $t1$ and $t2$ share an input place, but $\bullet t_1 = \bullet t_2 = \{p1\}$. Transitions $t5$ and $t6$ share an input place, but $\bullet t_5 = \bullet t_6 = \{p4, p5\}$.

The process model discovered using ProM (Figure 1) and Celonis (Figure 2) based on filtered SAP data is free-choice. Figure 3 shows the corresponding free-choice net.

[1] In a multiset elements may appear multiple times, e.g., $M = [p1, p2, p2, p2] = [p1, p2^3]$ is a multiset with four elements (three have the same value).

[2] $M_1 \subseteq M_2$ (inclusion), $M_1 \cup M_2$ (union),$M_1 \setminus M_2$ (difference) are defined for multisets in the usual way (i.e., taking into account the cardinalities. Sets are treated as multisets where all elements have cardinality 1.

C. Lucency

The notion of lucency was first introduced in [3]. A marked Petri net is *lucent* if there are no two different reachable markings enabling the same set of transitions, i.e., states are fully characterized by the transitions they enable.

Definition 10 (Lucent Petri nets): Let (N, M) be a marked Petri net. (N, M) is *lucent* if and only if for any $M_1, M_2 \in R(N, M)$: $en(N, M_1) = en(N, M_2)$ implies $M_1 = M_2$.

The marked Petri nets in Figures 3 and 5 are lucent, i.e., there are no two reachable markings that enable the same set of transitions. The marked Petri net in Figure 4 is not lucent. Markings $M_1 = [p2, p3, p4]$ and $M_2 = [p2, p3, p5]$ are both reachable and enable transitions $t3$ and $t4$.

Lucency is often a desirable property. Think, for example, of an information system that has a user interface showing what the user can do. In this setting, lucency implies that the offered actions fully determine the internal state and the system will behave consistently from the user's viewpoint. If the information system would not be lucent, the user could encounter situations where the set of offered actions is the same, but the behavior is very different. Another example is the worklist of a workflow management system that shows the workitems that can or should be executed. Lucency implies that the state of a case can be derived based on the workitems offered for it [6].

Characterizing the class of systems that are lucent is a foundational and also challenging question [3], [6], [7].

III. Free-choice Nets with Home Clusters

Workflow nets form a subclass of Petri nets starting with a source place *start* and ending with a sink place *end* [9]. The modeled workflow can be instantiated by putting tokens on the input place *start*. In the context of workflow nets, a correctness criterion called *soundness* has been defined [9]. A workflow net is sound if and only if the following three requirements are satisfied: for each case it is always still possible to reach the state which just marks place *end* (option to complete), if place *end* is marked all other places are empty for a given case (proper completion), and it should be possible to execute an arbitrary activity by following the appropriate route through the workflow net (no dead transitions) [9]. In [1], it was shown that soundness is decidable and can be translated into a liveness and boundedness problem, i.e., a workflow is sound if and only if the corresponding short-circuited net (i.e., the net where place *end* is connected to place *start*) is live and bounded. This can be checked in polynomial time for free-choice nets [1]. Figures 3 and 4 show two sound workflow nets. Figures 5 and 6 show free-choice nets that do not have a designated start and end place. Hence, soundness is not defined for these models.

A strongly-connected Petri net cannot be a workflow net. However, the lion's share of Petri net theory focuses on strongly-connected Petri nets. Therefore, [6] investigated a new subclass of Petri nets having a so-called *home cluster*.

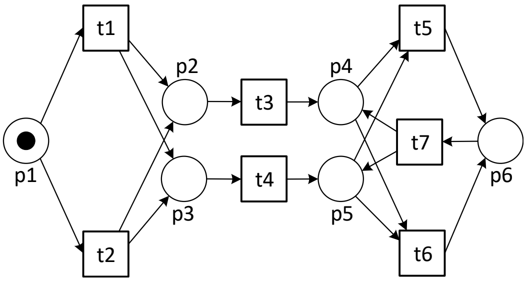


Fig. 6. A lucent free-choice net having two home clusters.

First, we define the notion of a *cluster*. A cluster is a maximal set of connected nodes, only considering arcs connecting places to transitions.

Definition 11 (Cluster): Let $N = (P, T, F)$ be a Petri net and $x \in P \cup T$. The *cluster* of node x, denoted $[x]_c$ is the smallest set such that (1) $x \in [x]_c$, (2) if $p \in [x]_c \cap P$, then $p\bullet \subseteq [x]_c$, and (3) if $t \in [x]_c \cap T$, then $\bullet t \subseteq [x]_c$. $[N]_c = \{[x]_c \mid x \in P \cup T\}$ is the set of clusters of N. $Mrk(C) = [p \in C \cap P]$ is the marking which only marks the places in C.

Figure 6 has five clusters: $[N]_c = \{\{p1, t1, t2\}, \{p2, t3\}, \{p3, p4\}, \{p4, p5, t5, t6\}, \{p6, t7\}\}$.

A home cluster is a cluster that serves as a "target" that can always be reached again. Hence, it can be seen as a generalization of soundness.

Definition 12 (Home Clusters): Let (N, M) be marked Petri net. C is a *home cluster* of (N, M) if and only if $C \in [N]_c$ (i.e., C is a cluster) and $Mrk(C)$ is a home marking of (N, M). If such a C exists, we say that (N, M) has a home cluster.

Figure 6 has two home clusters: $C_1 = \{p4, p5, t5, t6\}$ and $C_2 = \{p6, t7\}$.

Property 1 (Sound Workflow Nets Have A Home Cluster): Let (N, M) be a sound workflow net. (N, M) has a home cluster.

Also, all short-circulated sound workflow nets are guaranteed to have a home cluster. All marked Petri nets show thus far (i.e., Figures 3-6) have a home cluster. However, the nets in Figures 5 and 6 are not workflow nets.

Most of the results for Petri nets and in particular free-choice nets are defined for well-formed nets [11], [14], [15], [17], [19], [25], [29], [32]. Recall that a Petri net is well-formed if there exists a marking that is live and bounded. Some well-known properties of well-formed free-choice nets:

- A well-formed free-choice net is strongly connected.
- A bounded and strongly-connected marked free-choice net is live if and only if it is deadlock free.
- A marked free-choice net is live if and only if every proper siphon includes a marked trap.
- Well-formed free-choice nets are covered by P-components and T-components.
- Well-formedness can be decided in polynomial time for free-choice nets.
- Live and bounded free-choice nets have home markings.

Interestingly, marked free-choice nets having a home cluster do *not* need to be well-formed. Yet, free-choice nets having a home cluster have interesting properties as demonstrated in [6]. A surprising result is that free-choice nets having a home cluster are lucent.

Theorem 1 (Home Clusters Ensure Lucency [6]): Let (N, M) be a marked proper free-choice net having a home cluster. (N, M) is lucent.

The theorem can be used to show that the process models in Figures 3, 5, and 6 are lucent.

Theorem 1 is surprising since there are T-systems (i.e., marked graphs) that are live, bounded, safe, well-formed, and strongly connected that are not lucent. A proof of Theorem 1 is outside of the scope of this paper (see [6] for details). However, it is important to note that the proof does not rely on any of the classical results for well-formed nets. Instead, several new concepts are introduced, such as:

- *Expediting transitions* in a firing sequence of a free-choice net. As long as the order per cluster is maintained, transitions can fire earlier without causing any problems (e.g., deadlocks).
- The notion of *disentangled paths*, i.e., paths in the net that start and end with a place and do not contain elements that belong to the same cluster. A C-rooted disentangled path ends with a place in cluster C.
- A C-rooted disentangled path is *safe* if C is a home cluster. This implies that marked proper free-choice nets having a home cluster must be safe.
- The notion of *conflict-pairs*, i.e., a pair of markings such that no transition is enabled in both markings, but if a transition is enabled in one marking, the other marking must mark at least one of its input places.
- A marked proper free-choice net having a home cluster *cannot* have any conflict pairs.

These results make free-choice nets having a home cluster interesting candidate models in the context of BPM and process mining. However, as discussed next, there are also some limitations.

IV. Adding Non-Local Dependencies

Although many process discovery techniques return models that can be seen as free-choice and process modelers using BPMN are more-or-less forced to draw free-choice models, there are some limitations when using free-choice nets. Consider again the Petri net in Figure 4, which is not free-choice due to the places $p4$ and $p5$. The process model allows for the following four traces $L_1 = \{\langle t1, t3, t4, t5\rangle, \langle t1, t4, t3, t5\rangle, \langle t2, t3, t4, t6\rangle, \langle t2, t4, t3, t6\rangle\}$. Note that $t1$ is always followed by $t5$, and $t2$ is always followed by $t6$. In BPMN, we cannot express such dependencies (without resorting to data or other more advanced constructs). Ignoring the non-local dependencies represented by the places $p4$ and $p5$ leads to the BPMN model shown in Figure 7.

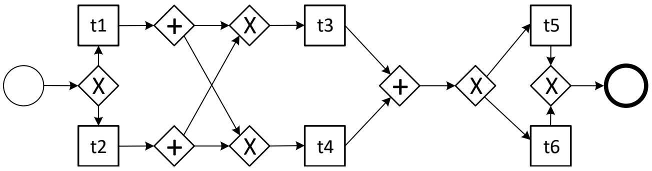


Fig. 7. A BPMN model that aims to describe the behavior in Figure 4 without local dependencies.

The corresponding free-choice net is shown in Figure 8. Both the BPMN model and the free-choice net allow for the following eight traces $L_2 = \{\langle t1, t3, t4, t5\rangle, \langle t1, t3, t4, t6\rangle, \langle t1, t4, t3, t5\rangle, \langle t1, t4, t3, t6\rangle, \langle t2, t3, t4, t5\rangle, \langle t2, t3, t4, t6\rangle, \langle t2, t4, t3, t5\rangle, \langle t2, t4, t3, t6\rangle\}$. Hence, the number of possibilities doubled.

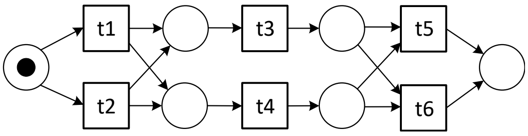


Fig. 8. The free-choice net corresponding to the BPMN model in Figure 7.

Most process discovery techniques will be unable to capture such non-local dependencies. Given an event log with only traces from L_1, most discovery techniques will produce a process model that allows for L_2. Some of the region-based process mining techniques can discover the process model allowing for only L_1. However, these techniques have many other problems: they tend to produce over-fitting models, cannot handle infrequent behavior, and are very time-consuming. Therefore, it may be better to first discover a *free-choice backbone model* that is then extended to make it more precise. Concretely, one can first discover a Petri net using the inductive mining approach and then add non-local dependencies. One can use, for example, a variant of the approach in [23] to add places. It is also possible to combine two types of arcs as in hybrid process models [10]. In [10], we use *hybrid Petri nets* and first discover a causal graph based on the event log. Based on different (threshold) parameters, we scan the event log for possible causalities. In the second phase, we try to learn places based on explicit quality criteria. Places added can be interpreted in a precise manner and have a guaranteed quality. Causal relations that cannot or should not be expressed in terms of places are added as sure or unsure arcs. A similar approach can be used for strongly correlating choices in a free-choice net.

There is also an interesting connection to the notion of *confusion*. Confusion is the phenomenon that the order of executing concurrent transitions may influence choices in the model. Here, we consider a simpler notion and consider a Petri net to be confusion-free when transitions that share an input place either cannot be both enabled or have the same set of input places.

Definition 13 (Confusion-Free): A marked Petri net (N, M) with $N = (P, T, F)$ is *confusion-free* if for any two transitions $t_1, t_2 \in T$ with $\bullet t_1 \cap \bullet t_2 \neq \emptyset$ and $\bullet t_1 \neq \bullet t_2$ there is no reachable marking $M' \in R(N, M)$ such that $\{t_1, t_2\} \subseteq en(N, M)$.

All models in this paper are confusion free. Note that free-choice nets are by definition confusion-free. An interesting question is to develop automatic conversions from models that are "almost free-choice".

Thus far concepts such as confusion-free, lucency, and home clusters have not been exploited in process mining using traditional event logs. In [5], an algorithm is presented assuming translucent event logs that explicitly show the enabling of activities. However, such event logs are rarely available.

V. Conclusion

In this paper, we discussed recent results in free-choice net theory and related these results to Business Process Management (BPM) in general and process mining in particular. Although most discovery techniques produce free-choice models, this property is rarely exploited explicitly. Assuming that the process model is a free-choice net with a home cluster, provides many valuable properties relevant for process discovery. As shown in this paper, such models are, for example, guaranteed to be lucent. This implies that there cannot be two states enabling the same set of activities. Also, disentangled paths rooted in a home cluster are safe, i.e., such paths cannot contain two tokens. The open question is how to exploit this in process mining.

We also discussed the need to add non-local dependencies. Such dependencies destroy elegant properties such as lucency. Hence, they can be seen as a secondary layer of annotations. For example, we can connect clusters that are strongly correlated. The goal is to make the process models more precise without overfitting the data or destroying the structure of the model.

Acknowledgment

The author thanks the Alexander von Humboldt (AvH) Stiftung for supporting our research.

References

[1] W.M.P. van der Aalst. The Application of Petri Nets to Workflow Management. *The Journal of Circuits, Systems and Computers*, 8(1):21–66, 1998.

[2] W.M.P. van der Aalst. *Process Mining: Data Science in Action*. Springer-Verlag, Berlin, 2016.

[3] W.M.P. van der Aalst. Markings in Perpetual Free-Choice Nets Are Fully Characterized by Their Enabled Transitions. In V. Khomenko and O. Roux, editors, *Applications and Theory of Petri Nets 2018*, volume 10877 of *Lecture Notes in Computer Science*, pages 315–336. Springer-Verlag, Berlin, 2018.

[4] W.M.P. van der Aalst. A Practitioner's Guide to Process Mining: Limitations of the Directly-Follows Graph. In *International Conference on Enterprise Information Systems (Centeris 2019)*, volume 164 of *Procedia Computer Science*, pages 321–328. Elsevier, 2019.

[5] W.M.P. van der Aalst. Lucent Process Models and Translucent Event Logs. *Fundamenta Informaticae*, 169(1-2):151–177, 2019.

[6] W.M.P. van der Aalst. Free-Choice Nets With Home Clusters Are Lucent. *Fundamenta Informaticae*, 181(4):273–302, 2021.

[7] W.M.P. van der Aalst. Reduction Using Induced Subnets to Systematically Prove Properties for Free-Choice Nets. In D. Buchs and J. Carmona, editors, *Applications and Theory of Petri Nets and Concurrency (PN 2021)*, volume 12734 of *Lecture Notes in Computer Science*, pages 208–229. Springer-Verlag, Berlin, 2021.

[8] W.M.P. van der Aalst and K.M. van Hee. *Workflow Management: Models, Methods, and Systems*. MIT Press, Cambridge, MA, 2002.

[9] W.M.P. van der Aalst, K.M. van Hee, A.H.M. ter Hofstede, N. Sidorova, H.M.W. Verbeek, M. Voorhoeve, and M.T. Wynn. Soundness of Workflow Nets: Classification, Decidability, and Analysis. *Formal Aspects of Computing*, 23(3):333–363, 2011.

[10] W.M.P. van der Aalst, R. De Masellis, C. Di Francescomarino, and C. Ghidini. Learning Hybrid Process Models From Events: Process Discovery Without Faking Confidence. In J. Carmona, G. Engels, and A. Kumar, editors, *International Conference on Business Process Management (BPM 2017)*, volume 10445 of *Lecture Notes in Computer Science*, pages 59–76. Springer-Verlag, Berlin, 2017.

[11] W.M.P. van der Aalst and C. Stahl. *Modeling Business Processes: A Petri Net Oriented Approach*. MIT Press, Cambridge, MA, 2011.

[12] W.M.P. van der Aalst, A.J.M.M. Weijters, and L. Maruster. Workflow Mining: Discovering Process Models from Event Logs. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1128–1142, 2004.

[13] A. Augusto, R. Conforti, M. Marlon, M. La Rosa, and A. Polyvyanyy. Split Miner: Automated Discovery of Accurate and Simple Business Process Models from Event Logs. *Knowledge Information Systems*, 59(2):251–284, May 2019.

[14] E. Best, J. Desel, and J. Esparza. Traps Characterize Home States in Free-Choice Systems. *Theoretical Computer Science*, 101:161–176, 1992.

[15] E. Best and H. Wimmel. Structure Theory of Petri Nets. In K. Jensen, W.M.P. van der Aalst, G. Balbo, M. Koutny, and K. Wolf, editors, *Transactions on Petri Nets and Other Models of Concurrency (ToPNoC VII)*, volume 7480 of *Lecture Notes in Computer Science*, pages 162–224. Springer-Verlag, Berlin, 2013.

[16] J. Carmona, B. van Dongen, A. Solti, and M. Weidlich. *Conformance Checking: Relating Processes and Models*. Springer-Verlag, Berlin, 2018.

[17] J. Desel and J. Esparza. *Free Choice Petri Nets*, volume 40 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, UK, 1995.

[18] C.A. Ellis and G.J. Nutt. *Computer Science and Office Information Systems*. Xerox, Palo Alto Research Center, 1979.

[19] J. Esparza. Reachability in Live and Safe Free-Choice Petri Nets is NP-Complete. *Theoretical Computer Science*, 198(1-2):211–224, 1998.

[20] S.J.J. Leemans, D. Fahland, and W.M.P. van der Aalst. Discovering Block-structured Process Models from Event Logs: A Constructive Approach. In J.M. Colom and J. Desel, editors, *Applications and Theory of Petri Nets 2013*, volume 7927 of *Lecture Notes in Computer Science*, pages 311–329. Springer-Verlag, Berlin, 2013.

[21] S.J.J. Leemans, D. Fahland, and W.M.P. van der Aalst. Discovering Block-Structured Process Models from Event Logs Containing Infrequent Behaviour. In N. Lohmann, M. Song, and P. Wohed, editors, *Business Process Management Workshops, International Workshop on Business Process Intelligence (BPI 2013)*, volume 171 of *Lecture Notes in Business Information Processing*, pages 66–78. Springer-Verlag, Berlin, 2014.

[22] S.J.J. Leemans, D. Fahland, and W.M.P. van der Aalst. Scalable Process Discovery and Conformance Checking. *Software and Systems Modeling*, 17(2):599–631, 2018.

[23] L. Mannel and W.M.P. van der Aalst. Finding Complex Process-Structures by Exploiting the Token-Game. In S. Donatelli and S. Haar, editors, *Applications and Theory of Petri Nets 2019*, volume 11522 of *Lecture Notes in Computer Science*, pages 258–278. Springer-Verlag, Berlin, 2019.

[24] M. Zur Muehlen and J. Recker. How Much Language Is Enough? Theoretical and Practical Use of the Business Process Modeling Notation. In Z. Bellahsene and M. Léonard, editors, *Proceedings of the 20th International Conference on Advanced Information Systems Engineering (CAiSE'08)*, volume 5074 of *Lecture Notes in Computer Science*, pages 465–479. Springer-Verlag, Berlin, 2008.

[25] T. Murata. Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE*, 77(4):541–580, April 1989.

[26] OMG. Business Process Model and Notation (BPMN). Object Management Group, formal/2011-01-03, 2011.

[27] C. Ouyang, M. Dumas, A.H.M. ter Hofstede, and W.M.P. van der Aalst. Pattern-Based Translation of BPMN Process Models to BPEL Web Services. *International Journal of Web Services Research*, 5(1):42–62, 2007.

[28] H.A. Reijers, I.T.P. Vanderfeesten, and W.M.P. van der Aalst. The Effectiveness of Workflow Management Systems: A Longitudinal Study. *International Journal of Information Management*, 36(1):126–141, 2016.

[29] W. Reisig. *Petri Nets: Modeling Techniques, Analysis, Methods, Case Studies*. Springer-Verlag, Berlin, 2013.

[30] A. Rozinat and W.M.P. van der Aalst. Conformance Checking of Processes Based on Monitoring Real Behavior. *Information Systems*, 33(1):64–95, 2008.

[31] F.W. Taylor. *The Principles of Scientific Management*. Harper and Bothers Publishers, New York, 1919.

[32] P.S. Thiagarajan and K. Voss. A Fresh Look at Free Choice Nets. *Information and Control*, 61(2):85–113, 1984.

[33] S.J. van Zelst, B.F. van Dongen, W.M.P. van der Aalst, and H.M.W Verbeek. Discovering Workflow Nets Using Integer Linear Programming. *Computing*, 100(5):529–556, 2018.

[34] M.D. Zisman. *Representation, Specification and Automation of Office Procedures*. PhD thesis, University of Pennsylvania, Warton School of Business, 1977.

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 17–26

DOI: 10.15439/2021F003
ISSN 2300-5963 ACSIS, Vol. 25

Combinatorial Testing of Context Aware Android Applications

Shraddha Piparia
University of North Texas
ShraddhaPiparia@my.unt.edu

David Adamo
Square, Inc.
dadamo@squareup.com

Renee Bryce, Hyunsook Do, Barrett Bryant
University of North Texas
{Renee.Bryce, Hyunsook.Do, Barrett.Bryant}@unt.edu

***Abstract*—Mobile devices such as smart phones and smart watches utilize apps that run in context aware environments and must respond to context changes such as changes in network connectivity, battery level, screen orientation, and more. The large number of GUI events and context events often complicate the testing process. This work expands the AutoDroid tool to automatically generate tests that are guided by PairwiseInterleaved coverage of GUI event and context event sequences. We systematically weave context and GUI events into testing using the pairwise interleaved algorithm. The results show that the pairwise interleaved algorithm achieves up to five times higher code coverage compared to a technique that generates test suites in a single predefined context (without interleaving context and GUI events), a technique that changes the context at the beginning of each test case (without interleaving context and GUI events), and Monkey-Context-GUI (which randomly chooses context and GUI events). Future work will expand this strategy to include more context variables and test emerging technologies such as IoT and autonomous vehicles.**

***Index Terms*—Context, Android, EDSs, Combinatorial**

I. Introduction

SMART phone applications are becoming increasingly prevalent nowadays with widespread adoption. Graphical User Interface (GUI) applications have more complex structures to process a wide variety of context data pertaining to different scenarios compared to conventional computer software. This makes it more difficult for testing since they tend to have a rich selection of features. Due to infinite event combinations and fragmentation of supported devices for GUI applications, it is challenging in terms of time and costs to test them. Smart phone context-aware applications are emerging in multiple domains after integration of small-scale micro electromechanical sensors. For example, the Uber [1] smart phone application uses location services to help passengers to purchase rides through their technology that is able to access a customer's location and nearby available vehicles and drivers. The app relies on user events to request service and context events, i.e., the Global Position System (GPS) reports their location. Some applications are based on advance machine learning algorithms such as CarSafe [2], a mobile phone application which uses driver behavior and road conditions to enable safe driving. This application uses the front camera to monitor the driver and rear camera to monitor road conditions when the smartphone is mounted on the car windscreen. The application also makes use of GPS, gyroscope, and accelerometer to infer the vehicle's movement and triggers alerts if dangerous driving behavior is detected. Incorporation of such context events further complicates the testing process.

Although mobile context aware applications has made our environment easier and intelligent, the complexity of such applications poses a challenge for testing. Researchers have proposed numerous techniques to simulate testing of context-aware mobile applications. Designing such methods is highly challenging because of the following reasons:

- Eco system fragmentation: Different mobile platforms often have unique architectures and features which make it difficult to generalize context simulation. Certain actions supported on one platform may not work for another platform which leads to platform-specific simulation of context events.
- Context heterogeneity: A wide variety of sensors are available which simulate context-aware behavior of mobile applications. Each sensor type may need a different mechanism suitable for simulation.
- Platform limitations: The Integrated Development Environments (IDEs) provide an easy simulation of Graphical User Interface (GUI) events on emulators but less support for simulation of context events.
- Budgets: Typically, mobile context-aware applications rely on various contextual input sources in addition to GUI events which make it infeasible to exhaustively test all input combinations.

This work presents a black-box approach to automatically generate test cases for context-sensitive Android applications. We expand the Autodroid tool [3] to focus on context-aware applications that are affected by four specific context variables: internet, power, battery, and screen orientation. The major components of Autodroid include the test builder, abstraction manager (which uses Appium to identify GUI events in different states of Application Under Test (AUT)), event selector, and event executor. The Autodroid tool is designed to automatically generate test suites with test cases of varying lengths. Each test case can be re-executed in isolation which makes it easier for testers to understand the scenarios and failure reproduction. Our context simulation provides a way to reduce time and cost and improves quality of testing. We use a pair-wise strategy to simulate context and GUI event combinations.We compare our test suites obtained from extended Autodroid compared our results with Monkey-Context-GUI tool. Results indicate that our pairwise strategy improves code coverage in comparison to Monkey-Context-GUI and NoContext for five applications and improves code coverage in comparison to ISContext for three out of five applications.

II. Background and Related Work

This section provides background information about context aware systems and testing.

A. Background

Context aware applications: A context event may affect the manner in which the mobile application responds to subsequent user interaction events. For example, when a button element is clicked, a mobile application that needs to respond by downloading a file from internet may exhibit different behaviors depending on whether or not the preceding context event indicated a switch into airplane mode. Context based applications may respond to a context event usually by some sort of change in state. This change in state may be physical and observable or may be logical and un-observable. For instance, a mobile application may reduce the rate at which it sends data over a network when battery levels go below a certain threshold. Android implements a BroadcastReceiver to listens for such context events. We formally define a context event in Definition 1.

Definition 1. A context event is a set of context variables. A context variable is a 2-tuple (c, a) where c is a context category and a is a context action.

WiFi	Battery	AC Power	Screen Orientation
Connected	Ok	Connected	Portrait
Disconnected	Low	Disconnected	Landscape
-	High	-	-

TABLE I: Combinatorial testing model with four context variables and different values for each variable

For example, Table I shows a combinatorial context model with four context variables (WiFi, battery, AC power and screen orientation) with two possible values for each context variable except battery which has three. We define a context event as combination of various values of these variables i.e. *c=* $\{$*WiFi = connected, Battery = high, Power = connected, ScreenOrientation = portrait*$\}$.

Android applications are Event Driven Systems (EDS) but unlike other traditional applications like web applications, they are more likely to sense and react to different kinds of events. These events could be generated by system itself or other applications. This section discusses previous studies which provide methods to test context-aware mobile applications.

B. Related Work

1) Online GUI testing: Many tools and techniques for automated GUI testing of mobile applications exist [4]–[12]. The majority of these tools do not consider context changes and potential interactions between context variables during test generation. Test suites generated in a single predefined context may explore only the subset of GUI states and code that is reachable in a predefined context.

Machiry et al. [5] consider an application as an event-driven program that primarily interacts with its environment using a sequence of events via the Android Framework, called Dynodroid. Dynodroid can observe the reaction of the application upon each event while employing it as a guide for the generation of the next event. In addition, Dynodroid permits the interleaving of events generated by machines (numerous inputs) with the events generated by humans (intelligent events). They examined the capability of Dynodroid on 50 open-source applications while comparing the results with the same obtained via manually exercising applications and Monkey. Their study demonstrated that while Dynodroid covered lesser Java source code when compared with human approach, Dynodroid was still better than Monkey. Furthermore, Monkey took approximately 20 times more events than Dynodroid.

Amalfitano et al. [13] discuss GUI test automation using algorithms that traverse GUIs through continuous interaction and exploration. The algorithms simultaneously define and run test cases during the execution of an application. They use a generalized parametric algorithm to extract key aspects of the testing techniques while delivering a framework that can be employed to define and compare these testing techniques. Autodroid uses online GUI testing techniques similar to the ones mentioned by Amalfitano et al. [13] and Dynodroid.

2) Context based GUI testing: Test cases that only consider GUI events reduces the likelihood of finding faults that are only triggered by changes in context. This has been acknowledged in mobile application testing research [5], [14]–[17]. Existing work has built mobile application GUI testing tools that consider context events and GUI events. One such tool is Dynodroid [5]. Dynodroid generates a sequence of GUI and context events that is then fed as input to a mobile application under test. Each individual event is added to the input sequence using a random selection strategy. Dynodroid does not offer a way to systematically reduce and explore the range of possible context and GUI event combinations.

Song et al. propose an alternative approach to testing context-sensitive behavior of mobile applications [17]. The demonstrated approach tests context-sensitive behavior by executing a test suite multiple times in different contexts. Executing a test suite in multiple contexts can result in a situation where other valid test cases become infeasible in contexts different from that in which they were generated. Also, the approach is not cost-effective since the number of test cases to be executed increases significantly with the number of contexts to be tested. Furthermore, the proposed approach does not consider the use or impact of context event sequences interleaved with GUI events. Adamsen et al. [18] and Majchrzak et al. [19] describe similar techniques for augmenting preexisting test suites with context information. Techniques that modify preexisting test suites or re-execute them in multiple contexts may cause test cases to become infeasible. This is because introduction of context events after test generation may cause the AUT's behavior to differ from the expected behavior in the preexisting test suites.

Amalfitano et al. [14] introduce interleaving context event sequences with GUI events during mobile application test generation. They adopt an event-patterns-based testing approach with sequences of context events that may be used to test a mobile application. The work demonstrates the benefits of using such event patterns for testing mobile applications. The experiments were carried out using a small set of manually and arbitrarily defined event patterns. Future work may create an event-pattern repository.

Griebe et al. [15] describe a model-based technique for automated testing of context-aware mobile applications. The technique requires manual creation of annotated UML Activity Diagrams that describe the behavior and context parameters of the AUT. The authors present an approach to incorporate sensor input for user acceptance tests [20]. This approach can generate sensor values as test cases by extending a UI testing tool, Calabash-Android [21]. To provide higher abstraction in test cases, this approach parses human language expressions (e.g. I shake the phone) to generate data using a mathematical model. The sensor data obtained can be used in test cases which are written in Gherkin language. Moran et al. [22] develop a tool called Crashscope that uses information from static analysis of source code to test contextual features in Android apps. Amalfitano et al. [14] propose a technique that requires manual specification of context event patterns that can be included in Android application test cases. The authors do not describe a systematic way to interleave context events with GUI events during test generation and do not consider potential interactions between context variables.

Liang et al. [23] present a cloud based service, CAIIPA, for testing context aware mobile applications. This study uses combinations of real world context events and improves crash and performance bug detection by 11.1x and 8.4x as compared to Monkey-Context-GUI-testing when evaluated on 265 Windows applications. While CAIIPA [23] utilizes real-world data for emulation of context events, the context events are limited to coarse-grained hardware parameters such as WiFi network, CPU device memory and sensor input and lacks on contexts such as screen orientation. Also, CAIIPA focuses only on Windows applications. Hu et al. [24] propose a cloud-based automation tool known as AppDoctor which injects actions such as network states, GUI gestures, intents, and the changes of device storage into an app to explore its possible executions.

Gomez et al. [25] describes a tool MoTiF, a crowd based based approach to reproduce context sensitive crashes for Android applications. Crash patterns are generated from subject applications and is modified for application under test. This technique can successfully reproduce crashes but crash patterns may miss some information which may not be generally applicable across applications. Ami et al. proposes MobiCoMonkey [26] which allows for contextual testing of Android applications. MobiCoMonkey utilizes the tool offered by Android SDK (Monkey) by associating context events with the tool. The context events are either introduced randomly by the tool or could be predefined by users. However, some GUI events are affected only by certain context events and randomly firing context events might miss on such dependencies which needs human intervention to be avoided. Also, MobiCoMonkey does not offer a systematic way to combine GUI and context events.

III. Context Modeling and Pairwise Event Selection Strategy

Our framework takes a context model as input that contains a set of context variables and possible values for each variable in order to generate a pairwise covering array. There are 24 possible value combinations for context events shown in Table I. It is possible to expand the combinatorial context model to include other variable (e.g. bluetooth, GPS, etc.). Changing the operating context of an application for exhaustive combination of context variables makes it computationally expensive since the number of combinations increases with the number of context variables. A t-way covering array can be used to model the operating context of an AUT. For a combinatorial model with k variables and v possible values for each variable, a *t-way covering array* $CA(N;t;k;v)$ has N rows and k columns such that each t-tuple occurs at least once within the rows, where t is the strength of interaction coverage [27].

ID	WiFi	Battery	AC Power	Screen Orientation
c_1	Disconnected	Low	Disconnected	Landscape
c_2	Connected	Low	Connected	Portrait
c_3	Disconnected	Okay	Connected	Landscape
c_4	Connected	Okay	Disconnected	Portrait
c_5	Disconnected	High	Disconnected	Portrait
c_6	Connected	High	Connected	Landscape

TABLE II: A 2-way covering array that defines six contexts

Table II shows a pairwise covering array for the combinatorial context model in Table I. To assess the functionality of the AUT, each operating context is modified as represented by the row of the covering array c_i. The CATDroid framework generates covering arrays to test interactions between a subset of context variables.

IV. Test suite construction framework

The CATDroid framework uses an *event extraction cycle* to iteratively select and execute events from the GUI of the application under test and to construct test cases one-event-at-a-time.

Figure 1 shows pseudo code for the test suite construction framework to automatically construct test suites for context sensitive Android apps. The algorithm requires input: (i) an Android Application Package (APK) file, (ii) context events

```
Input: android application package, AUT
Input: combinatorial context model, M
Input: initial context selection strategy, initialContextSelection
Input: event selection strategy, eventSelection
Input: test case termination criterion, terminationCriterion
Input: test suite completion criterion, completionCriterion
Output: test suite, $T$
1: $C_{all} \leftarrow$ generate context covering array from $M$
2: $T \leftarrow \phi$                                        ▷ test suite
3: repeat
4:     $t_i \leftarrow \phi$                                  ▷ test case
5:     $C_{curr} \leftarrow \boxed{\textit{InitialContextStrategy}(C_{all})}$
6:     add initial context event, $C_{curr}$ to test case $t_i$
7:     install and launch AUT, add launch event to $t_i$
8:     $s_{curr} \leftarrow$ initial GUI state
9:     while $\boxed{\textit{TerminationCriterion}}$ is not satisfied do
10:        $E_{all} \leftarrow$ GUI events in current GUI state $s_{curr}$
11:        $e_{sel} \leftarrow \boxed{\textit{EventSelectionStrategy}(S_{curr}, E_{all}, C_{all})}$
12:        execute $e_{sel}$
13:        $t_i \leftarrow t_i \cup \{e_{sel}\}$
14:        $s_{curr} \leftarrow$ current GUI state
15:    end while
16:    $T \leftarrow T \cup \{t_i\}$
17:    finalize test case (clear cache/SD card, uninstall app, etc.)
18: until $\boxed{\textit{CompletionCriterion}}$ is satisfied
```

Fig. 1: Pseudocode for the extended Autodroid framework (boxes indicate framework parameters)

```
Input: current GUI state, $s_{curr}$
Input: GUI events in current state, $E_{all}$
Input: covering array generated from pairwise combinations of
    context events, $C_{all}$
Input: set of covered context-GUI pairs, $Context-GUI-pairs$
Input: set of covered context-state pairs, $Context-state-pairs$
Output: GUI event or context event, $e_{sel}$
$c_{curr} \leftarrow$ get_current_emulator_context()
$e_{sel} \leftarrow$ select a GUI event $e_i \ \varepsilon \ E_{all}$ such that $(c_{curr}, e_i)$ is
    not in $Context - GUI - pairs$
if $e_{sel}$ is NULL then
    $Context - state - pairs \leftarrow (c_{curr}, s_{curr})$
    $c_{sel} \leftarrow$ select a context $c_i \ \varepsilon \ C_{all}$ such that $(c_i, s_{curr})$ is
    not in $Context - state - pairs$
    if $c_{sel}$ is not NULL then
        $e_{sel} \leftarrow c_{sel}$
        return $e_{sel}$
    else
        $e_{sel} \leftarrow$ select a GUI event $e_i \ \varepsilon \ E_{all}$ randomly
    end if
end if
$Context - GUI - pairs \leftarrow (c_{curr}, e_{sel})$
return $e_{sel}$
```

Fig. 2: Pseudocode for Pairwise Algorithm

needed for a combinatorial model, (iii) an initial context strategy, (iv) a test case termination criterion, and (v) test suite completion criterion. The test case termination criterion terminates the sequences of events either on the basis of the length of each sequence or probability. The test suite completion criterion may be predefined number of test cases or a fixed time. Lines 9-15 represent the event extraction cycle that incrementally constructs each test case. The framework requires specifications for several parameters (shown in boxes) to instantiate different test generation techniques. The algorithm includes four steps:

Step 1: Generate context covering array. Line 1 generates a covering array C using [28] from the combinatorial context model M specified as input. The covering array specifies a set of context events that will be used to test the AUT. Each context event specified in the covering array has a corresponding set of context variables that changes the operating context of the AUT. The generated set of context events is used to set initial context at the beginning of the test case as well as for the pairwise event selection strategy.

Step 2: Initialize test case. Lines 4-8 initialize each test case in the test suite. Line 4 creates an empty event sequence. Line 5 uses a predefined strategy to iterate over a context covering array C_{all} and selects a different context event at the beginning of each test case. Line 7 launches the AUT in the selected start context event and adds a launch event to the test case. Line 8 retrieves the initial GUI state of the AUT.

Step 3: Select and execute an event. The *EventSelectionStrategy* procedure call on line 11 uses a predefined strategy to select and execute a context event or GUI event in each iteration of the event extraction cycle (lines 9-15). Event execution often changes the GUI state of the AUT and/or the value of one or more context variables. This iterative event selection and execution incrementally constructs a test case that may include context events and GUI events. In each iteration of the event extraction cycle, the *EventSelectionStrategy* parameter specifies a strategy to choose (i) whether to execute a GUI event or context event and (ii) which particular event to execute given a set of available GUI events, and a context covering array. Figure 2 describes the technique to interleave context and GUI events. A single test case ends when the algorithm satisfies a predefined *TerminationCriterion*.

Step 4: Finalize test case. At the end of each test case, line 17 resets the state of the AUT and clears all data that may affect the outcome of subsequent test cases.

The algorithm generates multiple test cases until it satisfies the *CompletionCriterion* that specifies when the test suite is complete.

Pairwise event selection procedure. Figure 2 describes the *EventSelectionStrategy* to select either a context or a GUI event prior to adding an event in the test case. The algorithm requires the input: (i) current GUI state, (ii) GUI events available in current state and (iii) context covering array, (iv) the set of covered context-GUI event pairs, and (v) the set of covered context-state pairs. The event selection occurs at line 11 of Figure 1. The algorithm tracks coverage of context-GUI pairs and context-state pairs at the test suite level. Context-GUI pairs keep track of GUI events executed in a particular context which ensures that all possible GUI events are executed in a particular context before changing the context of the emulator. Context-state pairs keep track of context state change in a GUI state. We start by fetching the current context state of the emulator at line 1. A GUI event is selected from available events that has not been executed yet in the current context. If such a GUI event exists, the GUI event is selected and the context-GUI pair is marked as covered. Once all GUI events in a current context are covered, the algorithm changes the context by selecting a value from covering array indicated in lines 3-12 such that there is at least one GUI event in the current GUI state that has not yet been executed in the chosen context. This step occurs at line 5. All tie breaks are performed randomly. We consider a total of 6 contexts based on the 2-way covering array in Figure II and depending on the GUI events for the AUT, different context-to-GUI ratio may exist for different applications. When all context-state pairs and context-GUI pairs are covered in a particular GUI state, the algorithm randomly selects a GUI event.

App Name	**Installations**	**Version**	**Lines**	**Methods**	**Classes**
Diode	10,000+	1.3.2.2	7933	1134	209
BartRunner	100,000+	2.2.19	3644	750	135
Your Local Weather	5000+	5.6.4	15062	499	114
MovieDB	1000+	2.1.1	2719	319	81
Abcore	1000+	0.77	1215	197	46

TABLE III: Characteristics of selected Android apps

V. Experimental Study

This section presents results of an empirical study with five applications with characteristics mentioned in Table III.

A. Experimental Setup

We use Android 10.0 Pixel emulator with API 29 and generate 10 test suites with each technique for five applications chosen from F-droid [29] with a total of 200 test suites. These subject applications are instrumented with JaCoCo [30]. We use a fixed probability value of 0.05 to terminate a test case with a two second delay between execution of events for each test case so that the AUT can respond to each event. We set a fixed time budget of two hours for test suite completion. We compare the code coverage for test suites obtained from various techniques mentioned in Section V-B.

B. Variables and Measures

This section discusses the variables and metrics used in our experiments.

Independent Variable Our independent variable is test generation technique and we consider three controls (Monkey-Context-GUI, NoContext, and ISContext) and one heuristic (PairwiseInterleaved) as follows:

- The **NoContext** technique generates a test suite by executing GUI events without consideration for context changes using our CATDroid tool. We used the *NoContext* technique to construct test suites in a single context c = $\{$*WiFi=connected, Battery=OK, AC Power=connected, ScreenOrientation=Portrait*$\}$ that represents favorable operating conditions for the AUT.
- The **IterativeStartContext (ISContext)** technique selects a different context event at the beginning of each test case by iterating through the context covering array in a round-robin manner. After choosing a context, the technique makes a random selection among GUI events.
- The **PairwiseInterleaved** technique selects a different context event at the beginning of each test case by iterating through the context covering array in a round-robin manner using our CATDroid tool. It also systematically interleaves context events with GUI events by prioritizing the execution of GUI events in new contexts as described in Figure 2.
- **Monkey-Context-GUI** [31] takes a predefined number of events as input. It executes an action on the GUI application by performing clicks randomly on the screen coordinates regardless of whether the events are relevant to the application under test (AUT). Monkey-Context-GUI generates a single event sequence for each test suite. We configure Monkey-Context-GUI to generate multiple event sequences for each test suite of 120 minutes. For each application, we find the maximum event sequence length across test suites for NoContext, ISContext, and PairwiseInterleaved techniques and provide it as input to Monkey-Context-GUI. Monkey-Context-GUI executes context events randomly without knowledge of the GUI events for the AUT. We considered Monkey-Context-GUI as a baseline for evaluation of test suites obtained using CATDroid since it is one of the few existing tools that is compatible with recent versions of Android OS.

Dependent Variables We use the following code coverage metrics to investigate our research questions:

- **Line coverage** Line coverage measures the total number of covered source code statements.
- **Method coverage** Method coverage indicates whether a method was entered at all during execution.
- **Class coverage** Class coverage metric measures how many classes were executed by a test suite.

C. Research Questions

Our experiments address the following research questions:

RQ1: Does the PairwiseInterleaved technique increase line, method, and class coverage compared to Monkey-Context-GUI, NoContext, and ISContext?

Application	**Monkey-Context-GUI**			**NoContext**			**ISContext**			**PairwiseInterleaved**		
	Line Coverage	**Method Coverage**	**Class Coverage**	**Line Coverage**	**Method Coverage**	**Class Coverage**	**Line Coverage**	**Method Coverage**	**Class Coverage**	**Line Coverage**	**Method Coverage**	**Class Coverage**
Diode	6.29	11.56	9.28	32.44	43.81	38.42	33.33	45.37	40.08	**34.15**	**45.83**	**40.24**
Abcore	5.51	5.58	4.35	15.83	25.38	21.74	**58.63**	65.84	67.61	58.03	**65.94**	**68.48**
MovieDB	6.97	11.41	13.58	40.65	47.36	51.36	49.45	57.58	60.62	**52.71**	**59.56**	**63.58**
YourLocalWeather	4.73	8.31	16.66	9.04	15.31	27.92	9.19	15.44	**27.93**	**9.22**	**15.51**	26.74
BartRunner	10.31	12.43	20.96	**59.06**	**62.93**	**75.41**	58.09	61.60	**75.41**	51.94	55.77	70.89
Average	6.76	9.86	12.97	31.41	38.96	42.97	41.74	49.16	54.33	41.21	48.52	53.98

TABLE IV: Average code coverage for Monkey-Context-GUI, NoContext and PairwiseInterleaved test suites

RQ2: Do control techniques (NoContext, ISContext, and Monkey-Context-GUI) perform differently in terms of line, method, and class coverage?

D. Results and Analysis

Table IV shows the average line, method, and class coverage across ten runs of our techniques, Monkey-Context-GUI, NoContext, ISContext, and PairwiseInterleaved for each subject application. The values in bold indicate the highest values of line, method, and class coverage across all techniques for five subject applications. The bottom row shows the average values for each approach (i.e. Monkey-Context-GUI, NoContext, ISContext, and PairwiseInterleaved techniques) although the results vary across applications as discussed in Section V-E.

Application	**NoContext over Monkey-Context-GUI**		
	Line Coverage	**Method Coverage**	**Class Coverage**
Diode	5.16	3.78	4.13
Abcore	2.87	4.54	**4.99**
MovieDB	**5.83**	4.14	3.78
YourLocalWeather	*1.91*	*1.84*	*1.67*
BartRunner	5.72	**5.06**	3.59
Average	4.30	3.88	3.64

TABLE V: Ratio for NoContext over Monkey-Context-GUI

RQ1 Results: The PairwiseInterleaved technique shows twice as much line coverage when compared to Monkey-Context-GUI for the application *Your Local Weather*. On the other hand, the PairwiseInterleaved technique obtains significant improvements (up to ten times) in terms of line coverage for the application *Abcore*. Similarly, the test suites obtained from PairwiseInterleaved technique for four subject applications showed improvements in terms of line coverage over Monkey-Context-GUI. The method coverage and class coverage follows a similar pattern for the PairwiseInterleaved technique when compared to Monkey-Context-GUI.

For the application *Bart Runner*, the NoContext and ISContext techniques outperformed the PairwiseInterleaved technique in terms of line coverage. The applications *Your Local Weather* and *Diode* show similar line coverage for test suites obtained from the NoContext, ISContext, and PairwiseInterleaved technique. However, for the application *Movie DB*, a considerable improvement in line coverage was observed for the PairwiseInterleaved technique over the NoContext and ISContext techniques. We notice similar behavior for both techniques in terms of method and class coverage.

Our PairwiseInterleaved technique often achieved 6.1 times (absolute difference of 34.45%), 4.9 times (absolute difference of 38.66%), and 4.2 times (absolute difference of 41.02%) higher line, method, and class coverage in comparison to Monkey-Context-GUI. It also achieved 1.31 times (absolute difference of 9.8%), 1.25 times (absolute difference of 9.7%), and 1.26 times (absolute difference of 11.01%) higher line, method, and class coverage in comparison to NoContext. However, the PairwiseInterleaved technique does not show an improvement over ISContext on an average. This is because the average value does not compare the impact of individual application. We notice that PairwiseInterleaved technique indicates better or comparable results for three out of five applications with an average of 1.37% improvement in terms of line, method, and class coverage.

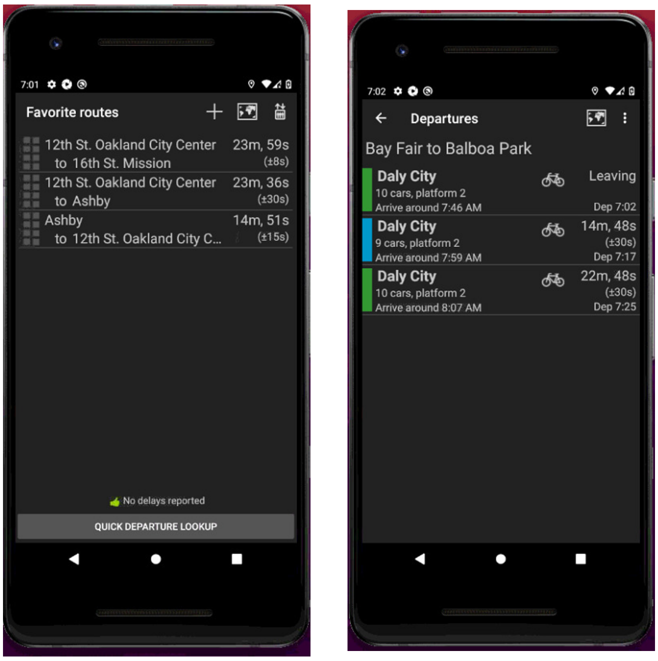
RQ2 Results: Table IV shows that line, method, and class coverage for the ISContext technique offers little improvement in comparison to the NoContext technique for *Your Local Weather* and *Diode* applications. The improvement in line coverage for ISContext over the NoContext technique was highest for the application *Abcore* (by a factor of 3.7). The application *MovieDB* showed an improvement in line and method coverage for ISContext over NoContext by a factor of 1.21. The class coverage showed 1.18 times improvement for *MovieDB*. The application *Bart Runner* performs better in terms of line and method coverage for NoContext when compared to ISContext by a small margin (by 1% and 1.3% respectively). The class coverage is same for both techniques. On an average, the ISContext technique achieved 1.32 times (10.32%), 1.26 times (10.2%), and 1.26 times (11.35%) higher line, method, and class coverage in comparison to NoContext.

Table IV indicates that ISContext achieves a higher line, method, and class coverage than Monkey-Context-GUI for all five applications. The application *Abcore* achieves 10.64 times line coverage, 11.79 times method coverage, and 15.54 times class coverage for ISContext technique over Monkey-Context-GUI. This is the highest improvement obtained between any two techniques across all subject applications. The applications *BartRunner* and *Diode* show approximately 5 times improvement in line coverage for ISContext in comparison to Monkey-Context-GUI. The method and class coverage follow a similar pattern. The ISContext technique for the *MovieDB* application show 7 times line coverage, 5 times method coverage and 4.4 times class coverage when compared to Monkey-Context-GUI. The application *Your Local Weather* shows an improvement for ISContext over Monkey-Context-GUI by a factor of 1.94 times for line coverage, 1.85 times for method coverage, and 1.7 times for class coverage. On an average, ISContext technique achieved 6.17 times (absolute difference of 41.74%) higher line coverage, 5 times (absolute difference of 49.16%) higher method coverage, and 4.2 times (absolute difference of 54.33%) higher class coverage when compared to Monkey-Context-GUI.

Table IV shows that line, method, and class coverage for the NoContext technique showed improvements over the Monkey-Context-GUI for *Your Local Weather* application (by a factor of more than 1.5). Likewise, the improvement in line coverage for the NoContext technique was highest for the application *Movie DB* whereas *Bart Runner* and *Abcore* showed the highest improvement in method and class coverage, respectively. An average, the NoContext technique achieved 4.6 times (absolute difference of 24.64%), 4 times (absolute difference of 29.10%), and 3.3 times (absolute difference of 30%) higher line, method, and class coverage in comparison to Monkey-Context-GUI.

Application	ISContext over NoContext		
	Line Coverage	Method Coverage	Class Coverage
Diode	1.03	1.04	1.04
Abcore	**3.70**	**2.59**	**3.11**
MovieDB	1.22	1.22	1.18
YourLocalWeather	1.02	1.01	*1.00*
BartRunner	*0.98*	*0.98*	*1.00*
Average	1.33	1.26	1.26

TABLE VI: Ratio for ISContext over NoContext



12th St. Oakland City Center to 16th St. Mission 23m, 59s (±8s)
12th St. Oakland City Center to Ashby 23m, 36s (±30s)
Ashby to 12th St. Oakland City C... 14m, 51s (±15s)

Bay Fair to Balboa Park
Daly City 10 cars, platform 2 Arrive around 7:46 AM Leaving Dep 7:02
Daly City 9 cars, platform 2 Arrive around 7:59 AM 14m, 48s (±30s) Dep 7:17
Daly City 10 cars, platform 2 Arrive around 8:07 AM 22m, 48s (±30s) Dep 7:25

(a) Bart Runner Routes

(b) Bart Runner Departures

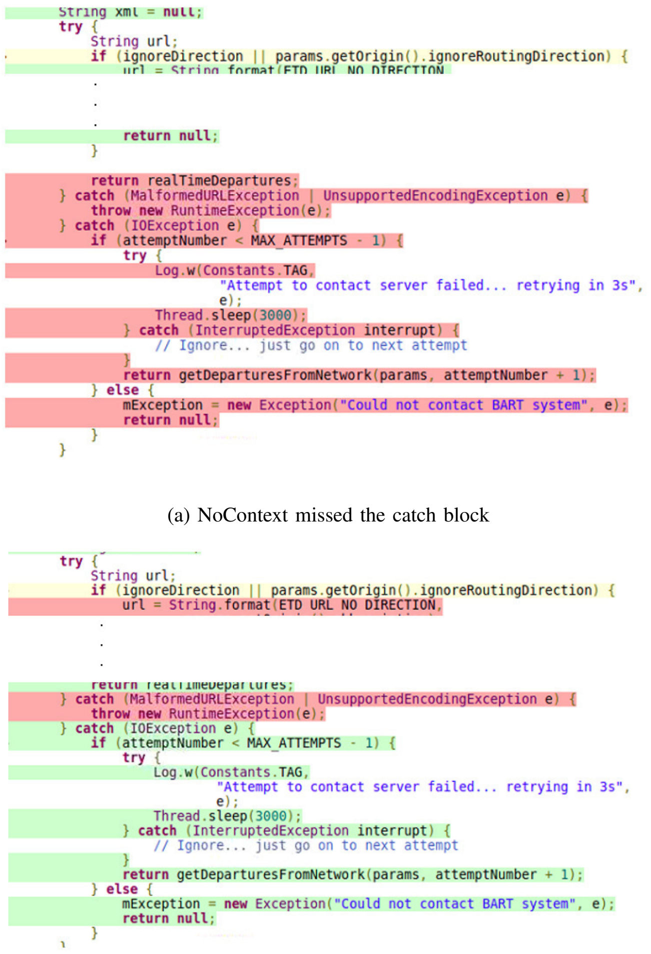
Fig. 3: The Bart Runner application

E. Discussion and Implications

To understand the performance improvement of NoContext over Monkey-Context-GUI, we calculated their ratios (NoContext:Monkey-Context-GUI) of coverage for all subject applications, as shown in Table V. The numbers in bold indicate the highest ratio and the italicized numbers indicate the lowest ratio obtained for various code coverage. The bottom row shows the average improvement of ratios of NoContext technique when compared to Monkey-Context-GUI across all subject applications. On an average, NoContext achieves 4.3 times line coverage, 3.88 times method coverage, and 3.64 times class coverage with different values across applications. We can see that the application *Movie DB* shows the highest ratio of line coverage across all applications. In addition, the apps *Bart Runner* and *Abcore* show improvement in method and class coverage, respectively. *Your Local Weather* also shows an improvement, although not as significant as other four applications. Although, Monkey-Context-GUI has an advantage for *Your Local Weather* application over extended Autodroid because Monkey-Context-GUI can add a location by choosing coordinates from the map component. Although, it still did not add a location due to its random nature. This analyses show that the NoContext technique outperforms our baseline, Monkey-Context-GUI, for all subject applications even though Monkey-Context-GUI considers execution of context events. This is because Monkey-Context-GUI does not have any information regarding the events in AUT. The random clicks performed by Monkey-Context-GUI sometimes does not lead to any action performed and hence the tool is not able to explore the application after a certain point. It is important to have information about the GUI events and states of the AUT to enable proper testing of Android applications.

We calculate the ratios of code coverage for ISContext over NoContext in Table VI. The bold and italicized values highlight the highest and lowest ratios. The bottom row shows the average. The only app for which ISContext does not perform better than NoContext is *Bart Runner*. The app makes use of internet to download train schedules but the huge number of GUI events overshadows context events and hence results in a low coverage for ISContext by approx 1%. This indicates that for this application, including context events could result in decreased code coverage and is not an ideal candidate for context-aware testing. *Your Local Weather* and *Diode* show a marginal improvement of ISContext over NoContext. This is because of limited dependency of these apps on context events. The application *Movie DB* and *Abcore* shows substantial improvement (approx. 9% and 41%) in line coverage for ISContext over NoContext.

The PairwiseInterleaved technique does not perform well for *BartRunner* when compared to NoContext and ISContext. *Bart Runner* is a scheduling application for trains in the US. It allows users to enter their most traveled routes and provides real-time list of upcoming departures. This application depends only on the three context variables; internet, wake locker, and alarm, but has numerous GUI events related to train routes as shown in Figure 3a and departures in Figure 3b. This leads to a high GUI to context events ratio which results in low coverage for our PairwiseInterleaved technique. For ISContext technique, the results are similar since the overhead of context over GUI events is not too much. However, both PairwiseInterleaved and ISContext techniques were able to cover some of the branches due to the absence of internet, which is not covered by NoContext algorithm across all test suites. Figure 4a shows the missing catch block for the NoContext technique which is covered by our Pairwise as



```
        String xml = null;
        try {
            String url;
            if (ignoreDirection || params.getOrigin().ignoreRoutingDirection) {
                url = String.format(ETD_URL_NO_DIRECTION,
            .
            .
            .
                return null;
            }

            return realTimeDepartures;
        } catch (MalformedURLException | UnsupportedEncodingException e) {
            throw new RuntimeException(e);
        } catch (IOException e) {
            if (attemptNumber < MAX_ATTEMPTS - 1) {
                try {
                    Log.w(Constants.TAG,
                            "Attempt to contact server failed... retrying in 3s",
                            e);
                    Thread.sleep(3000);
                } catch (InterruptedException interrupt) {
                    // Ignore... just go on to next attempt
                }
                return getDeparturesFromNetwork(params, attemptNumber + 1);
            } else {
                mException = new Exception("Could not contact BART system", e);
                return null;
            }
        }
```

(a) NoContext missed the catch block

```
        try {
            String url;
            if (ignoreDirection || params.getOrigin().ignoreRoutingDirection) {
                url = String.format(ETD_URL_NO_DIRECTION,
             .
             .
             .
            return realTimeDepartures;
        } catch (MalformedURLException | UnsupportedEncodingException e) {
            throw new RuntimeException(e);
        } catch (IOException e) {
            if (attemptNumber < MAX_ATTEMPTS - 1) {
                try {
                    Log.w(Constants.TAG,
                            "Attempt to contact server failed... retrying in 3s",
                            e);
                    Thread.sleep(3000);
                } catch (InterruptedException interrupt) {
                    // Ignore... just go on to next attempt
                }
                return getDeparturesFromNetwork(params, attemptNumber + 1);
            } else {
                mException = new Exception("Could not contact BART system", e);
                return null;
            }
        }
```

(b) PairwiseInterleaved and ISContext covers the catch block covered due to absence of Internet

Fig. 4: Code snippet for BartRunner for NoContext, ISContext, and PairwiseInterleaved techniques

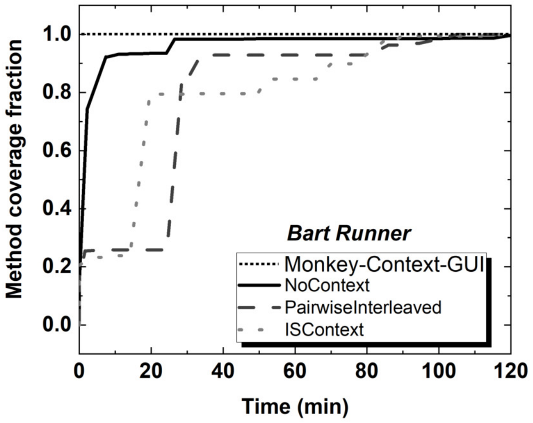


Fig. 5: Rate of method coverage for test suite with highest method coverage obtained for the application *Bart Runner*

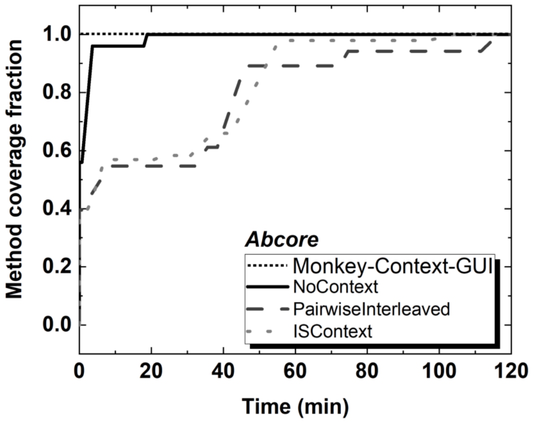


Fig. 6: Rate of method coverage for test suite with highest method coverage obtained for the application *Abcore*

well as ISContext techniques as shown in Figure 4b. *Bart Runner* uses two context events, alarm and wake locker, which were excluded from this study. Since applications are sensitive to specific context variables, it is important to have this information in advance and integrate those context variables in test case generation.

Abcore runs a BitCoin core node on the Android device. This app starts by downloading data from WiFi and may not work well on 3G or 4G network. *Abcore* has very limited number of GUI events and high context events which lead to only 15% line coverage in case of NoContext. The coverage is significantly improved (by a factor of 3.7) when context is manipulated at the beginning of each test case as evident by values of the technique ISContext. Also, *Abcore* is the only application which includes explicit broadcasts for all four context variables (internet, power, battery, and WiFi). Given that NoContext technique explores the functionality of the AUT only under favorable conditions (esp. internet connected) for all GUI states, low code coverage is obtained due to lack in consideration of different values for these context variables. Due to these reasons, PairwiseInterleaved technique offers a large increase over the NoContext technique. The PairwiseInterleaved technique gave slightly lesser line coverage but slightly higher method and class coverage when compared to ISContext.

We also investigate the method coverage rate of Monkey-Context-GUI, NoContext, ISContext, and PairwiseInterleaved techniques for *Bart Runner* and *Abcore* applications. We plot the method coverage graph for the application *Bart Runner*. We chose method coverage here but line and class coverage follows a similar pattern. Figure 5 shows the method coverage rate for the test suite with maximum method coverage for Monkey-Context-GUI, NoContext, ISContext, and PairwiseInterleaved techniques. Here, the abscissa indicates the time and ordinate indicates the fraction of method coverage. This fraction is obtained by normalizing the method coverage value by its maximum value. We plot this value with respect to time to obtain the method coverage rate. Similarly, we plot the method coverage graphs for the application *Abcore*. Figure 6 shows the rate of method coverage for test suites with maximum coverage. Monkey-Context-GUI reaches its maximum value from the beginning. NoContext reaches its maximum value rapidly for maximum coverage test suite when compared to ISCOntext and PairwiseInterleaved for both of these applications. The Monkey-Context-GUI and NoContext technique does not explore the AUT after it reaches its maximum value at a very early stage. The ISContext takes more time to reach its maximum value as compared to Monkey and NoContext but PairwiseInterleaved technique explores the AUT towards the later stage as well. Here, we show the graph of the test suite with maximum coverage but the test suite with minimum coverage also shows similar behavior.

Next, we analyze the results of the *Diode* and *Movie DB* applications due to their similar performances for PairwiseInterleaved technique when compared to ISContext and NoContext. *Diode* is a third party application which allows users to narrow the search for a particular reddit topic pertaining to a chosen theme. It features several choices for users which lead to numerous GUI events. *Movie DB* is an online application to explore database for movies and TV shows. NoContext explores approximately 40% of the *Movie DB* application and 32% of the *Diode* due to its huge number of GUI events. The ISContext technique shows 9% improvement in line coverage for the *Movie DB* when compared to NoContext. Similarly, ISContext shows 1% improvement in line coverage for the *Diode* application when compared to NoContext. This indicates that it is important to manipulate the context of the AUT during test generation. The PairwiseInterleaved technique slightly outperforms NoContext and ISContext for both of these applications. One reason for the favorable performance of ISContext and PairwiseInterleaved is that deeply nested GUI actions (i.e. actions far beyond the first screen) are affected by context. This is especially in case of the availability of an internet connection. So, despite the significant ratio of GUI events to context events there is a lot of variability in the outcome of a GUI event depending on internet availability or some other context. This indicates the importance of execution of GUI events in multiple contexts to test context-sensitive behavior. The overall code coverage for *Diode* application is also affected by the large number of topics available on reddit from which this application acquires its data.

The NoContext, ISContext, and PairwiseInterleaved techniques for *Your Local Weather* give similar results and show small improvement over Monkey-Context-GUI. *Your Local Weather* is a weather application that uses data network, WiFi, and GPS to show the weather of the current location. The application integrates a map component for users to specify a location. There are various reasons for the low code coverage for this application. The large size of this application in terms of lines of code made it difficult to explore the application within the allocated time frame. The map component is missing from our tool and hence the application is not able to detect a location. None of the techniques fully explored the application which resulted in overall lack of coverage. Including GPS to detect a location may improve code coverage for this application. Furthermore, the application considers the context variable $DEVICE\_BOOT\_COMPLETE$ to autostart the application after the device is finished booting, which is excluded from our study.

F. Implications for Mobile Application Testing

GUI test case generation algorithms produce a sequence of events as test cases [5], [32], [33]. The behavior of a sequence of GUI events can vary depending on the current operating context of a mobile device. It is important to also generate events that manipulate the operating context of the device in addition to exercising the AUT using GUI events.

Figure 1 provides an algorithm to automatically generate test cases with interleaved sequences of context and GUI events. The algorithm uses a combinatorial model to define different contexts and uses the algorithm mentioned in Figure 2 to determine next likely context event. This helps to limit the decision space at each point of adding a context event to the test case.

Context-aware test cases have the potential to expose context-driven behavior that may otherwise go untested without context events. For instance, tapping a ‘download’ button in a mobile app may exhibit varying behavior depending on whether or not an internet connection is available. Context-aware test cases may even discover interactions between the operating context of the device and GUI events. However, the potential benefits of context-aware test cases depend on the nature of the AUT. If the AUT does not use the Internet in any way, changing the connectivity context of the device is unlikely to expose new behavior.

VI. Threats to Validity

This work applies the context driven testing strategies to five different applications that have different characteristics in terms of size, relevant context events, number of screens, and behaviors. The results may differ for applications with different characteristics. Another threat is the random nature of our techniques. To control this threat, we performed ten runs for test suite generation and reported the average values. One challenge for research in context-sensitive mobile app testing is the lack of reliable emulators for context events. This hinders automated generation of context-aware tests for mobile applications. The context variables in this study were limited to events possible to simulate in an Android emulator. A larger set of context variables and a different schedule for context event insertion may help achieve better results.

VII. Conclusion and Future Work

Smart phone applications are EDS which also react to context events that may cause its behavior to change. Context events may alter the operating context of an application under test which makes it important to generate tests that manipulate the operating context of the AUT. This work provides a context aware automated testing framework (CATDroid) for automatic generation of test suites. We use our framework to instantiate multiple test case generation techniques that compares test case generation techniques with and without manipulating the context of AUT. The PairwiseInterleaved technique achieves higher line coverage up to a factor of six when compared to Monkey-Context-GUI, up to a 1.3 times increment in line coverage compared to a technique that generates test suites in a single predefined context, and achieves similar code coverage when compared to ISContext across all five subject applications. Our results indicate that the benefit of manipulating operating context and interleaving context events with GUI events depend on the characteristics of the AUT. Future work will explore a broader set of applications and context events and the impact of higher interaction strength coverage of context and GUI events.

References

[1] Uber, "Uber- earn money by driving or get a ride now," 2019, retrieved Feb 25, 2020 from https://www.uber.com.

[2] C.-W. You, M. Montes-de Oca, T. J. Bao, N. D. Lane, H. Lu, G. Cardone, L. Torresani, and A. T. Campbell, "Carsafe: a driver safety app that detects dangerous driving behavior using dual-cameras on smartphones," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, 2012, pp. 671–672.

[3] D. Adamo, D. Nurmuradov, S. Piparia, and R. Bryce, "Combinatorial-based event sequence testing of android applications," *Information and Software Technology*, vol. 99, pp. 98–117, 2018.

[4] D. Amalfitano, A. R. Fasolino, P. Tramontana, B. D. Ta, and A. M. Memon, "MobiGUITAR: Automated model-based testing of mobile apps," *IEEE Software*, vol. 32, no. 5, pp. 53–59, 2015.

[5] A. Machiry, R. Tahiliani, and M. Naik, "Dynodroid: An input generation system for android apps," in *Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering*. ACM, 2013, pp. 224–234.

[6] I. C. Morgado and A. C. Paiva, "The iMPAcT tool: Testing UI patterns on mobile applications," in *2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2015, pp. 876–881.

[7] N. Mirzaei, J. Garcia, H. Bagheri, A. Sadeghi, and S. Malek, "Reducing combinatorics in GUI testing of android applications," in *Proceedings of the 38th International Conference on Software Engineering*. ACM, 2016, pp. 559–570.

[8] D. Amalfitano, N. Amatucci, A. M. Memon, P. Tramontana, and A. R. Fasolino, "A general framework for comparing automatic testing techniques of android mobile apps," *Journal of Systems and Software*, vol. 125, pp. 322–343, 2017.

[9] K. Mao, M. Harman, and Y. Jia, "Sapienz: Multi-objective automated testing for android applications," in *Proceedings of the 25th International Symposium on Software Testing and Analysis*. ACM, 2016, pp. 94–105.

[10] R. Michaels, D. Adamo, and R. Bryce, "Combinatorial-based event sequences for reduction of android test suites," in *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*, 2020, pp. 0598–0605.

[11] R. Michaels, M. K. Khan, and R. Bryce, "Mobile test suite generation via combinatorial sequences," in *ITNG 2021 18th International Conference on Information Technology-New Generations*, S. Latifi, Ed. Cham: Springer International Publishing, 2021, pp. 273–279.

[12] S. Piparia, M. K. Khan, and R. Bryce, "Discovery of real world context event patterns for smartphone devices using conditional random fields," in *ITNG 2021 18th International Conference on Information Technology-New Generations*, S. Latifi, Ed. Cham: Springer International Publishing, 2021, pp. 221–227.

[13] D. Amalfitano, N. Amatucci, A. R. Fasolino, and P. Tramontana, "A Conceptual Framework for the Comparison of Fully Automated GUI Testing Techniques," in *2015 30th IEEE/ACM International Conference on Automated Software Engineering Workshop (ASEW)*, 2015, pp. 50–57.

[14] D. Amalfitano, A. R. Fasolino, P. Tramontana, and N. Amatucci, "Considering context events in event-based testing of mobile applications," in *2013 IEEE Sixth International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. IEEE, 2013, pp. 126–133.

[15] T. Griebe and V. Gruhn, "A model-based approach to test automation for context-aware mobile applications," in *Proceedings of the 29th Annual ACM Symposium on Applied Computing*. ACM, 2014, pp. 420–427.

[16] Z. Liu, X. Gao, and X. Long, "Adaptive random testing of mobile application," in *2010 2nd International Conference on Computer Engineering and Technology (ICCET)*, vol. 2. IEEE, 2010, pp. V2–297.

[17] K. Song, A. R. Han, S. Jeong, and S. Cha, "Generating various contexts from permissions for testing android applications," in *27th International Conference on Software Engineering and Knowledge Engineering (SEKE)*, 2015, pp. 87–92.

[18] C. Q. Adamsen, G. Mezzetti, and A. Møller, "Systematic execution of android test suites in adverse conditions," in *Proceedings of the 2015 International Symposium on Software Testing and Analysis*. ACM, 2015, pp. 83–93.

[19] T. A. Majchrzak and M. Schulte, "Context-dependent testing of applications for mobile devices," *Open Journal of Web Technologies (OJWT)*, vol. 2, no. 1, pp. 27–39, 2015.

[20] T. Griebe, M. Hesenius, and V. Gruhn, "Towards automated UI-tests for sensor-based mobile applications," in *International Conference on Intelligent Software Methodologies, Tools, and Techniques*. Springer, 2015, pp. 3–17.

[21] Uber, "Calabash-android," 2019, retrieved Feb 25, 2020 from https://github.com/calabash.

[22] K. Moran, M. Linares-Vásquez, C. Bernal-Cárdenas, C. Vendome, and D. Poshyvanyk, "Automatically discovering, reporting and reproducing android application crashes," in *2016 IEEE International Conference on Software Testing, Verification and Validation (ICST)*. IEEE, 2016, pp. 33–44.

[23] C.-J. M. Liang, N. D. Lane, N. Brouwers, L. Zhang, B. F. Karlsson, H. Liu, Y. Liu, J. Tang, X. Shan, R. Chandra, and F. Zhao, "Caiipa: Automated large-scale mobile app testing through contextual fuzzing," in *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '14. New York, NY, USA: ACM, 2014, pp. 519–530. [Online]. Available: http://doi.acm.org/10.1145/2639108.2639131

[24] G. Hu, X. Yuan, Y. Tang, and J. Yang, "Efficiently, effectively detecting mobile app bugs with appdoctor," in *Proceedings of the Ninth European Conference on Computer Systems*, 2014, pp. 1–15.

[25] M. Gómez, R. Rouvoy, B. Adams, and L. Seinturier, "Reproducing context-sensitive crashes of mobile apps using crowdsourced monitoring," in *Proceedings of the International Conference on Mobile Software Engineering and Systems*, ser. MOBILESoft '16. New York, NY, USA: ACM, 2016, pp. 88–99. [Online]. Available: http://doi.acm.org/10.1145/2897073.2897088

[26] A. S. Ami, M. M. Hasan, M. R. Rahman, and K. Sakib, "Mobicomonkey - context testing of android apps," in *2018 IEEE/ACM 5th International Conference on Mobile Software Engineering and Systems (MOBILESoft)*, May 2018, pp. 76–79.

[27] R. C. Bryce and C. J. Colbourn, "Prioritized interaction testing for pairwise coverage with seeding and constraints," *Information and Software Technology*, vol. 48, no. 10, pp. 960–970, 2006.

[28] H. Tsuyoshi. (2021) Pairwise combinatosila package. https://github.com/thombashi/allpairspy. (Accessed: 25-05-2021).

[29] F-Droid, "F-droid: Free and open source android app repository," http://f-droid.org, 2017, (Accessed: 26-02-2021).

[30] Mountainminds GmbH, "EclEmma: JaCoCo java code coverage library," http://www.eclemma.org/jacoco/, 2017, (Accessed: 26-02-2021).

[31] Google, "UI/application exerciser monkey," 2017, retrieved May 10, 2021 from https://developer.android.com/studio/test/monkey.html.

[32] A. M. Memon, "Developing testing techniques for event-driven pervasive computing applications," in *Proceedings of The OOPSLA 2004 workshop on Building Software for Pervasive Computing (BSPC 2004)*, 2004.

[33] A. Memon, "An event-flow model of gui-based applications for testing," *Software testing, verification and reliability*, vol. 17, no. 3, pp. 137–157, 2007.

DOI: 10.15439/2021F004
ISSN 2300-5963 ACSIS, Vol. 25

Machine Learning and High-Performance Computing Hybrid Systems, a New Way of Performance Acceleration in Engineering and Scientific Applications

Pawel Gepner
Warsaw Technical University
ul. Narbutta 86, 02-524 Warszawa, Poland
email: pawel.gepner@pw.edu.pl

***Abstract*—Machine learning is one of the hottest topics in IT industry as well as in academia. Some of the IT leaders and scientists believe that this is going to totally revolutionise the industry. This transformation is happening on both fronts, one is the application and software paradigm, the other is at the hardware and system level. At the same time, the High-Performance Computing segment is striving to achieve the level of Exascale performance. It is not debatable that to meet such level of performance and keep the cost of system and power consumption on reasonable level is not a trivial task. In this article, we try to look at a potential solution to these problems and discuss a new approach to building systems and software to meet these challenges and the growing needs of the computing power for HPC systems on the one hand, but also be ready for a new type of workload including Artificial Intelligence type of applications.**

***Index Terms*—Machine Learning, High-Performance Computing, Exascale performance, HPC systems, IPU**

I. Introduction

TODAY'S High-Performance Computing systems are build out of thousands of nodes containing a couple of CPUs and one-or-many accelerators, mostly GPUs, onto the same node. All theses nodes are connected using dedicated fabric to make a cluster, the most common and universal supercomputer of the present day. Clusters definitely dominate the HPC market and occupy 92% of all the systems ranked on the TOP 500 list [1]. From an architecture stand point one-such- node of the system represents a hybrid architecture, including CPU processor and accelerating unit. Basically we find the major CPU architecture commonly used for HPC are Intel/AMD x86, Arm, and Power. According the latest Top 500 list more than 28% of the systems utilize GPUs [1], also on this segment we have a choice between, Nvidia and AMD, as well as, soon to be third player — Intel with their GPU architecture. Integrating hundreds or even thousands of heterogeneous nodes together requires special interconnect technology and advanced topology for the network part. Today we have basically two solution scenarios: InfiniBand with low latency and standard Ethernet with all its advantages and disadvantages. Unfortunately, both new technologies, such as OPA or BXI, which could become a potential alternative, have not achieved significant market penetration and adoption to be considered real players.

From a programming perspective, such solutions are not trivial, we need to properly handle heterogeneous infrastructure and be fully aware of the hybrid structure of code. Effective use and management of the offload-acceleration parts and utilizing CPU portions of the code is not a simple task and requires experienced and knowledgeable developers.

There is no doubt that in the case of large Exascale systems, homogeneous systems based on CPUs only are unable to provide the required performance in a reasonable size and do not destroy the power budget. It seems that the first few Exascale systems, we are going see in the next couple of years, will be based on a heterogeneous architecture - CPUs plus GPUs, although in diverse combinations of vendors for these components.

Artificial Intelligence-AI and Machine Learning-ML systems have been discussed for decades, but limited access to large data sets , lack of relevant computing architectures and available systems able to execute the AI workloads have restrained AI developments, until the last couple of years.

Artificial intelligence systems are based on one of the combinations of the following types of devices in conjunction with a Central Processor Unit:

- GPU – Graphics Processing Units.
- FPGA – Field Programmable Gate Arrays.
- ASIC – Application Specific Integrated Circuits.

Many of today's AI systems only use modern, efficient multi-core processors to solve ML tasks. Processor vendors are trying to support this segment by optimizing the architecture, extending instruction sets to best address common ML workloads, but also by adding a new data format and special precision for this type of computation.

Despite these efforts by CPU manufacturers, processor-based systems are not considered the most effective solution for processing AI-oriented tasks, especially in the training phase, and dedicated hardware has earned a reputation as the best solution for this type of applications.

Solutions such as GPU, FPGA or specialized ASICs such as Tensor Processing Units-TPU from Google, Inferentia from Amazon, Gaudi and Goya from Habana Labs, Grayskull from Tenstorrent, SambaNova AI Chip, Cerebras Wafer Scale Engine or a special dedicated Intelligent Processor Unit-IPU from Graphcore are gaining great recognition and a reputation for being the best solution for ML workloads.

Initially, the GPU was designed to accelerate multidimensional data processing in computer graphics applications. The GPU consists of thousands of small cores designed to work independently. On the one hand ensures intensive processing, but at the same time requires advanced management of the memory subsystem and operating on the subset of data on which it runs. A GPU can perform complex computational operations for computer graphics such as texture mapping, resizing and cropping images, rotating and flipping, translation and filtering, and the dedicated memory makes these operations fast and efficient.

The HPC community realized the possibilities and benefits of using the GPU to accelerate linear algebra calculations long before the AI researchers started using them to solve machine learning problems. The GPU has been found to benefit wherever we are able to take advantage of the natural parallelism of algorithms and accelerate them by parallel execution. At the same time, from specialized chips dedicated to graphics solutions they became fully programmable graphics processors. Today they are still specialized parallel processors, but also highly programmable and the spectrum of implementation has grown significantly.

Deep neural network calculations are based on a similar type of operation as linear algebra calculations, so the natural consequence of this fact is the use of GPUs to solve deep learning problems. GPUs are throughput processors and can provide high throughput and be an effective solution for accelerating HPC and machine learning type of systems.

Unlike GPUs, FPGAs did not make a stunning career in the HPC market but for Machine Learning specially in inference it is very interesting alternative for already trained model. FPGAs are an array of programmable logic blocks and memory, connected via a hierarchy of reconfigurable interconnects that allow the blocks to be wired together to perform specific complex tasks. Since the FPGA is not a processor, it cannot execute the program stored in memory, but it can perfectly execute the code for which it has been configured. FPGAs are inherently parallel, so they're a perfectly matching parallel computing capabilities of machine learning models. To configure and store the executable program in to the chip and make it operational we use a fully parallel hardware description language - HDL. Results show that FPGA provides superior performance/Watt over CPU and GPU because FPGA's on-chip extracting fine-grained parallelisms and matches the specifics of the machine learning code.

Specialized ASICs for AI solutions, like all customised chips, have their own specific requirements optimized for AI, which radically accelerate the calculations demanded by ML algorithms. They are based on massive parallel operations that are performed simultaneously, instead of sequentially, to achieve the required performance. Dedicated AI chips require low precision computation in a way that AI algorithms implement them effectively, this approach reduces precision but speeds up execution and reduces the number of transistors required for the same computation. Customised AI ASICs are characterized by super-efficient memory access, where they store all of the data necessary for the correct implementation of the algorithm, therefore, with the growth of models, the memory of such chips also evolves over time. Many companies have brought AI specialised chips to the market and number of new solutions increases every day. Different types of AI chips are useful for different tasks some of them were developed to do training others to do inference others for both domains.

In this article, we propose a hybrid system architecture capable of solving both HPC and AI problems. Such a solution is utilising classic HPC platforms and dedicated AI chip systems, which is particularly important in the context of the challenges of the new types of Exascale systems. We will discuss not only the implications at the chip level, but also the system approach along with the necessary modification of algorithms and software, as well as the performance. The approach of the heterogeneous system presented in the article will allow us to propose a system capable of accelerating the most complex simulation problems. The construction of such a heterogeneous system is based on customized AI chips and systems utilizing Graphcore's Intelligent Processor Unit-IPU for executing a new hybrid algorithm acceleration scenario. Of course on the market we can find other ASICs and dedicated AI type of chips but IPU appears to be the most mature and universal solution available on the market today, with a solid and comprehensive software ecosystem.

This paper is organized as follows. Section 2 presents related works. Section 3 is devoted to the introduction of the Intelligent Processor Unit. It explains architectural details of IPU and describes the way it works and the principles of it's programming model. In Section 4, we concentrate on the details of the proposed system architecture, discuss the benefits and challenges. Section 5 discusses the programming approach and the new way algorithms are tuned to the architecture that enables the use of a hybridization approach. In Section 6, we present the conclusions of the conducted architectural proposal.

II. Related Works

Over the last few years, there have been many attempts at improving the architecture as well as many discussions on development directions of hardware technology, as well as attempts to use the existing High Performance Computing technology in Machine Learning frameworks and artificial intelligence systems [2]. The convergence of HPC and artificial intelligence [2], [3] offers a promising approach to major performance improvements. As classic HPC simulations are reaching the limits of their progress and slowing down due to the stagnation of Moore's law that has led to the proliferation of various accelerator architectures. These architectures are still evolving, resulting in very costly, if not harmful, modification of the scientific codes that must be optimised to bring out the last marginal gains of parallelism and efficiency. In many application areas, the integration of traditional HPC approaches with machine learning methods is perhaps the greatest promise to overcome these barriers.

The need to increase performance is the clue to the international efforts behind the Exascale supercomputer projects. Exascale initiatives in the US, Japan and Europe exploring the possible integration of ML with large-scale computing is a very promising way to achieve high performance. There is undoubtedly a close relationship between machine learning and high performance computing as machine learning algorithms are based on the same basic linear algebra operations that are used in HPC. The question that arises is how to efficiently use the existing HPC infrastructure for ML applications and can ML code be incorporated into HPC simulation and is this approach optimal?

Interesting attempts to answer these questions can be found in works such as Jeff Dean's ”Machine Learning for Systems and Systems for Machine Learning” [4] and Satoshi Matsuoka's "Convergence of AI and HPC" [5]. However, the most advanced categorization of such systems, and even the development of a specific taxonomy, can be found in Geoffrey Fox's et al. "Learning Everywhere: Pervasive Machine Learning for Effective High-Performance Computation" [2]. The authors attempted to classify a new category of system and the type of calculations associated with them, they proposed a method for their hierarchy, and even defined some performance metrics .They distinguished classical performance as measured by Flops or benchmark results from effective performance, which is achieved by combining learning with simulation and delivering increased performance as seen by the user. This is absolutely crucial in the cases where there is a combination of machine learning and the components of a traditional HPC simulation where classical benchmarks are not representative at all.

On the basis of the proposed classification, several clearly different approaches to the construction of systems using machine learning and HPC solutions have been identified [2]. Basically, two categories of systems are distinguished: HPCforML and MLforHPC:

- HPCforML: Utilising HPC to run and enhance ML performance, or using HPC simulations to train ML algorithms.
- MLforHPC: Deploy ML to enhance HPC applications and systems.

Of course, the proposed split divides the system based on particular type of calculations. If ML and AI is the major target, then we can utilize the HPC installations for acceleration of AI type of code, but if we have a HPC traditional problem and AI and ML seems to be a good way for acceleration run time than, the system belongs to the second class [3]. This is absolutely clear that the main categories of systems have their subclasses and they are defined as follows:

- HPCrunsML: Using HPC to perform ML type of workloads with highest degree of effective using the HPC acceleration capabilities [2], [3].
- SimulationTrainedML: Using HPC simulations to train ML algorithms, which are then used to understand experimental data [2], [3].
- MLautotuning: Using ML to configure autotune HPC simulations. MLautotuning can also be used for simulation mesh sizes and in big data problems for configuring databases and complex systems like Hadoop and Spark [2], [3].
- MLafterHPC: ML analyzing results of HPC simulation [2], [3].
- MLaroundHPC: Using ML for learning from simulations and creating learned solvers replacing classic HPC wrappers [2], [3].
- MLControl: Using ML to control of experiments and simulation run on HPC system [2], [3].

There are many research groups working on all of these HPC and AI hybridization categories, and most of the work is just starting and still in an early stage of development, but by far the most attractive in terms of performance and potential is MLaroundHPC. Some groups have already very im- pressive results with MLaroundHPC especially in high energy physics, materials science, weather simulation, epi- demic forecasting, tissue and cellular simulations, nanoscale and biomolecular simulation, computational fluid dynamics simulation and many others [3].

Another aspect often discussed in the context of combining AI and HPC is the question of measuring the performance of such solutions. In the classic model of measuring the efficiency of HPC systems, the situation is quite obvious "faster is better" which in turn prompts the acceleration of individual work units. For hybrid systems, the ML component should be included, and this requires both hierarchical (vertical) and horizontal (multitasking) parallelism [6].

Much effort has been put into developing a new type of hybrid systems that can scale to very large sizes and bring performance benefits to a whole new level that is impossible to achieve based on classical simulation and HPC. The proposed hybrid approach in this paper incorporating AI and HPC components takes into account these new trends and proposes the use of solutions that have already proved their usefulness in AI solutions.

III. Intelligent Processor Unit

Bulk Synchronous Parallel - BSP model is the foundation for the architectural assumptions of the IPU Colossus MK2 GC200 processor and the Poplar programming model. Valiant proposed the Bulk Synchronous Parallel model as a parallel computing abstraction scheme that facilitates the expression of parallel algorithms, helps design large paralleled systems and makes it easier to analyse the performance they achieve while running [7].

This model, proposed in the 1980s, is a parallel counterpart to the Von Neumann model's for sequential computing, and allows exemplification structure of parallel algorithms and can be used for performance characteristic measurement [8].

BSP is a parallel computing model that IPUs use to organize data processing and exchange operations. The way the BSP model arranges computation is based on sequential 3-steps phases [8]. This 3-steps phase is constituted of:

- local computation phase - every process performs computation that operates exclusively on local memory. No communication between processes occurs in this phase.
- communication phase - processes exchange data and each process may send a message to each looked-for destination counterpart. No computation occurs in this phase. Processes can use the communication phase not only to send each other intermediate computation results, but also to request and (at a later communications stage) receive data from remote memories. This mechanism allows each process to use other local memory as a remote memory and to ultimately access the entire aggregate system memory as one larger store.
- • barrier synchronization phase - no process continues to the next phase till all processes have reached the barrier. Neither computation nor communication occurs in this phase except for that strictly required by the barrier itself.

The BSP model is ideally suited to describe any parallel algorithms of any complexity and allowing software developers to specify processes in terms of graph vertices that compute on local data. Input operands are fetched to each process by the run-time system before the computation begins, in the communication phase related to the previous cycle [8].

The hardware-assisted programming model not only enforces phase separation but also ensures the coupling of both entities. Since the IPU cores can only access directly the local memories on the chips, this naturally imposes local constraints on the computational phase. The hardware-implemented all-to-all memory exchange mechanism provides native support and enforces control and exchange during the communication and synchronization barriers phases [7].

From the hardware definition IPU is distributed memory, massively parallel, multiple-instruction, multiple-data (MIMD) devices. Each IPU has 1472 cores, each with its own on-chip 624KiB SRAM memory per core, combination of core and associated on-chip memory is named a tile. With 1472 tiles the IPU has just under 900 MB of memory in total. This local memory is the only memory directly accessible by tile instructions. It is used for both the code and data used by that tile. There is no shared memory access between tiles. The tile uses a contiguous unsigned 21-bit address space, beginning at address 0x0. The effect of accessing unpopulated memory addresses is undefined. Memory parity errors can occur when data is read from memory, for example, by a load instruction or an instruction fetch. A parity error detected in a fetched instruction prevents the execution of that instruction. Tiles cannot directly access each other's' memory but can communicate via message passing using an all-to-all high bandwidth exchange (theoretical 8 TB/s). The memory has very low-latency (6 cycles) and ultra-high bandwidth (theoretical 47.5 TB/s). The whole chip is built on the budget of 59.4 billion transistors and using the TSMC 7nm manufacturing process [9]. Fig. 1 shows simplified schematic of IPU die.

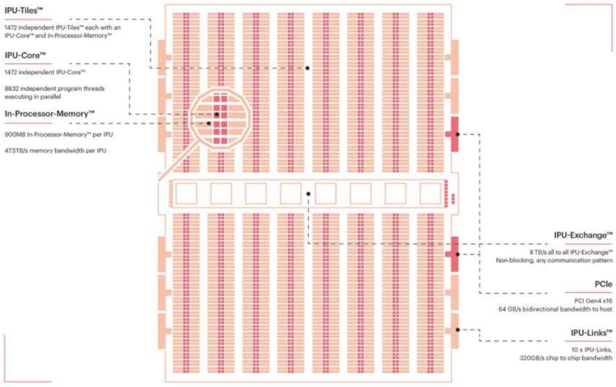

IPU-Tiles™
1472 independent IPU-Tiles™ each with an IPU-Core™ and In-Processor-Memory™
IPU-Core™
1472 independent IPU-Core™
8832 independent program threads executing in parallel
In-Processor-Memory™
900MB In-Processor-Memory™ per IPU
47.5TB/s memory bandwidth per IPU
IPU-Exchange™
8 TB/s all to all IPU-Exchange™
Non-blocking, any communication pattern
PCIe
PCI Gen4 x16
64 GB/s bidirectional bandwidth to host
IPU-Links™
10 x IPU-Links,
320GB/s chip to chip bandwidth

Fig. 1. Simplified version of IPU die

The IPU is specifically designed for machine learning type of computation, and the tile Instruction Set Architecture-ISA[15] includes focussed hardware elements such as Accumulating Matrix Product-AMP units and Slim Convolution Units-SLICs which allow up to 64 multiply-add instructions to complete per clock cycle. There are also hardware support instructions for random number generation and some selected, generally used in machine learning, transcendental operations. The IPU supports 32-bit single-precision floating point FP32-IEEE, as well as FP16-IEEE 16-bit half-precision floating point numbers of data format with hardware stochastic rounding support. Every tile runs 6 hardware execution threads in a time-sliced round-robin schedule, allowing instruction and memory latency to be hidden. With this mechanism, most instructions, including memory access and vectorised floating-point operations, complete within one thread cycle (6 clock cycles). Every thread represents a truly independent program, there is no restriction that threads run in groups executing the same program in lockstep, and no requirement that memory accesses are coalesced to achieve the high SRAM bandwidth [9].

All these architecture principals have been carefully selected to entirely support the machine learning specific workloads but at the same time they make the IPU one of the most powerful devices on the market. Table 1 shows comparison of existing CPUs and GPUs available on the market. This table clearly illustrates that IPU has a significant advantage in terms of the available resources but also theoretical performance versus other devices uses for machine learning applications.

The programming interface to access the IPU is the Graphcore Poplar programming framework. Poplar is a graph programming environment that essentially extends the functionality of C ++ by transforming it into the IPU operation model and is based on three concepts:

- Vertexes are the programs which execute on individual tiles. Vertices in Poplar are subclasses of the Vertex class. They each have a compute method that is run on the

TABLE I
CHARACTERISTICS OF CPUS, GPUS AND IPU

Architecture	Memory	Capacity	Frequency (GHz)	Bandwidth (GB/s)	FP32-TFLOPS	FP16-TFLOPS
Graphcore IPU	SRAM	900 MB	1.325	47500	61	245
Intel, Xeon 8380	L1/L2/L3/DRAM	48KB/1.25MB/60MB/4TB	2.3	7048/5424/1927/225	5.8	11,6
AMD 7742 (Rome)	L1/L2/L3/DRAM	4MB/32MB/256MB/4TB	2.25	190	4.7	9,4
Nvidia GPU-A10	L1/L2/HBM-2	128KB/6MB/24GB	0.885	600.2	31.2	31.24
Nvidia GPU-A100	L1/L2/HBM-2	192KB/40MB/40GB	0.765	1555	19.5	77.97
Nvidia GPU-V100	L1/L2/HBM-2	128KB/6MB/16GB	1.312	897	16.4	31.33
AMD GPU-MI-100	L1/L2/HBM-2	16KB/8MB/32GB	1	1229	23.07	184.6

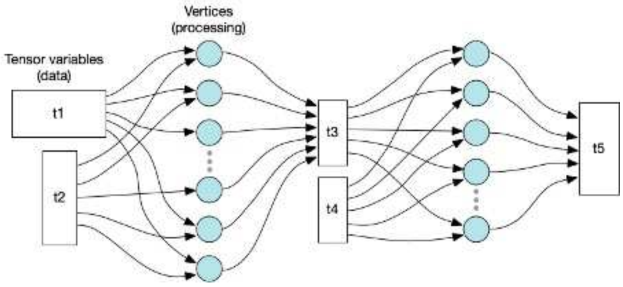


Fig. 2. IPU Computation Graphs concept

tile and returns a bool value. Vertexes are defining an interface of inputs and outputs which later allows them to be wired into the Computation Graph. The function performed by a vertex can be anything from a simple arithmetic operation to reshaping tensor data operation or performing a complicated code.

- Computation graph defines the input/output relationship between variables and operations. Poplar provides functionality for constructing, compiling, and serialising the computation graph.
- Control programs administrate arguments, select IPU devices, control the execution of the graph operations.

Fig 2 shows the concept of the IPU computation graphs which defines the input/output relationship between variables and operations.

The graph is made up of tensor variable (variables in the graph), compute tasks (vertices) and edges that connect them. Data is stored in the graph in fixed size multi-dimensional tensors, a vertex is a specific piece of work to be carried out and the edges determine which variable elements are processed by the vertex. A vertex can connect to a single element or a range of elements. Each vertex is associated with a codelet - piece of code that defines the inputs, outputs and internal state of a vertex. Codelet is implemented in standard C++11 [10]. Example 3.1 shows simple example of the Adder vertex (vertex that adds two numbers).

Example 3.1 (C++): The Adder vertex:

```
#include <poplar/Vertex.hpp>

using namespace poplar;
class AdderVertex : public Vertex {
public:
  Input<float> x;
  Input<float> y;
  Output<float> sum;

  bool compute() {
    *sum = x + y;
    return true;
  }
};
```

The final element that brings all the elements together is the control program, it organizes the selection of devices, loads compiled graphs into the IPU and executes graph programs. An important part of this is the mapping of data transfers between the IPU and the host, memory structures, and initiating transfers. Once the program is deployed, all the code and data structures required to run the program reside in the IPU's distributed memory [10]. The control programs run in order to execute the appropriate vertices.

IV. HYBRID SYSTEM ARCHITECTURE

The next generation of AI systems promises to accelerate computer vision, speech recognition, machine translation systems and increasingly impact our lives. Realizing this promise we need AI systems that can compute massively increasing amounts of data and do it in realistically short time. In the same time the size of the HPC systems is increasing but the level of utilization does not necessarily evolve in the same direction. The most extensive and sophisticated of today's HPC models are so computationally expensive that researchers need to replace parts of the model with approximation mechanisms that transfer accuracy to speed, only to obtain a simulation that can be performed at feasible scale and resolution. It turns out that ML models are universal and accurate approximations, so one direction of research is to use accurate but slow numeric code to generate training data for the ML model that can then be implemented on a large scale, more efficiently, and maybe even more accurately. To make this transformation successful we need the specialized hardware dedicated for ML framework from one hand but should not jeopardise the HPC requirements. Due to different architectural characteristics and the large number of system parameter configurations (such as, the number of threads, thread affinity, workload partitioning between multi-core processors of the host and the accelerating devices), achieving a good workload distribution that results with optimal performance for HPC 64 bits and ML 32 bit and 16-bit workloads is not a trivial task. An optimal system configuration that results with the highest throughput and performance for HPC may not necessarily be the most effective solution for AI type of solvers. Moreover, the optimal system configuration for AI workloads is likely to change for the different types of applications, sizes of input problems, and available resources that we have in HPC. Taking all these elements into account the proposed approach of hybridisation the system architecture appears to be a creditable solutions.

Suggested systems contain two parts AI and HPC dedicated portion but from the networking, storage and orchestration, administration point of view it is a single instance. This type of the approach provides an optimized solution for the monolithically type of HPC code which is running on the HPC section of the system without any limitation, as well as the AI portion can be utilized only when we have ML workload. The most complicated scenario appears for the hybrid type of code when they have HPC section involved in the preparation phase for the data generating process for AI optimized solver. These new architectures will aim not only to improve the performance, but to simplify the development of the next generation of AI and HPC hybrid applications by providing rich libraries of modules that are easily composable.

Many institutions have already started investigation and development of hybrid systems for example, the University of California, Berkeley Firebox project or Lawrence Livermore National Laboratory - LLNL with their two heterogenic systems. The LLNL integrated AI-specific systems one from Cerebras and one from SambaNova into two existing LLNL HPC systems (Lassen (23 petaflops) and Corona (10 petaflops)) to achieve system level heterogeneity [11]. Many other works have been carried out and many centres are experimenting with different setups and configurations. Most of today's research is concerned with the allocation of resources and the interaction between different types of resources. One should also consider what is the proper ratio of these different types of resources and what should be a reasonable size of the systems between the HPC and AI components. A particularly important question is the need to provide disaggregated network bandwidth that will be sufficient and network latency low enough to tie the two domains together and meet the needs of each task. Fig. 3 shows the simplified version of the system with separated HPC classic system and ML section. The HPC section is CPU or CPU plus accelerator based. This part is dedicated for typical HPC simulation workload. Second section is ML system responsible for AI type of workloads. Both systems are connected via unified network e.g. Ethernet and they are utilizing the storage subsystem which can be shared simultaneously or as the aggregated bandwidth subsystem for hybrid type of workloads.

The proposed architecture can utilize any type of HPC system based on the existing infrastructure as we see e.g., in LLNL or completely new specially built system. For the machine learning component, the proposed system is based on Graphcore architecture for a couple of reasons. The performance of the IPU itself and architecture customisation for ML work- loads already discussed in the previous section of the article, also because of the unique approach to the architecture of the IPU systems, based on disaggregated approach for the platform level integration and scalability.

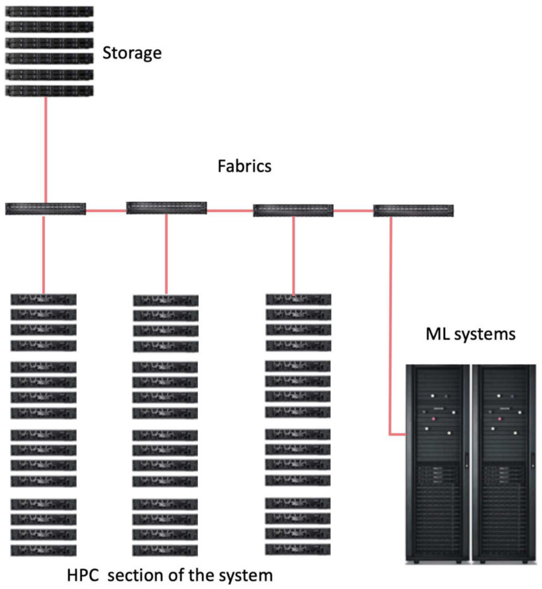


Fig. 3. Schematic of the hybrid system

Graphcore IPU-M2000 system is basically a 1U server utilizing 4 IPUs, gateway chip which connects IPUs into compute domain, provides access to the DRAM, two 100Gbps IPU-Fabric Links, a PCIe slot for standard Smart NICs, two 1GbE Open BMC management interfaces, and access to an M.2 slot. Fig.4 shows the block diagram of the IPU-M2000 system. The host system accesses IPU-M2000 platform over 100Gb Ethernet with ROCE (RDMA over Converged Ethernet) with very low-latency access. Such an implementation based on Ethernet avoids the bottlenecks and costs of PCIe connectors and PCIe switches and enables a flexible host CPU to accelerators combination and provides the scaling from single IPU-M2000 system to massive supercomputer scale including 64000 IPUs, all networked over standard networking at lower cost and much more flexibility than using e.g., InfiniBand [12].

The IPU-Fabric is a totally new scale-out fabric designed from the ground up to support the needs of machine intelligence communication. The IPU–Fabric is natively integrated into the IPU processors and IPU-M2000 system. A key difference between IPU-Fabric and other proprietary fabrics are the usage of Compiled Communication and Bulk Synchronous Protocol, both these elements provide deterministic communication behaviour. Every IPU has dedicated IPU-Links providing 64GB/s of bidirectional bandwidth and an aggregate

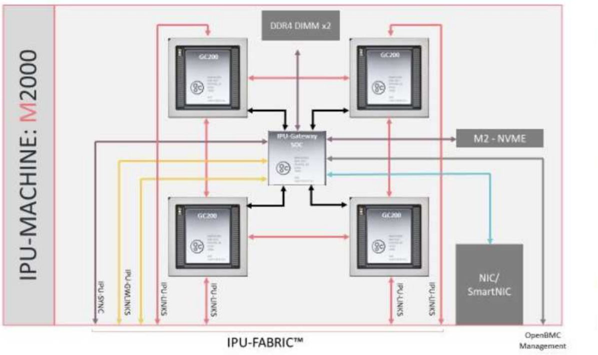


Fig. 4. Schematic and building block of IPU-M2000 Machine

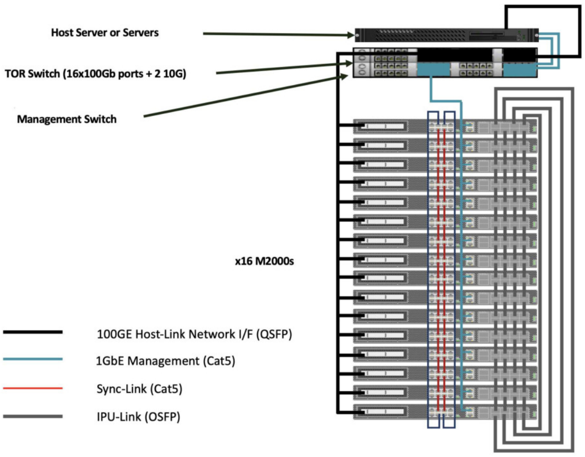


Fig. 5. IPU-POD64 configuration

bandwidth per chip of 320 GB/s. Each IPU- M2000 has 8 external IPU-Links for intra-rack scale out using OSFP copper cables. The intra-rack configuration called IPU- POD64 contains 16 of IPU-M2000's connected into a single instance with 2D ring topology utilizing IPU-Links. Host-Link connectivity is provided from the GW through a PCIe NIC or SmartNIC card. Fig. 5 shows the IPU-POD64 configuration [12].

For inter-rack scale out the IPU-GW provides 100GbE ports that tunnel the IPU-Link protocol over regular Ethernet, theses links are named as IPU-GW-Links. Physically every IPU-M2000 system has 2 QSFP ports that support both optical transceivers and copper cables for rack-to-rack connectivity. Each IPU-GW-Link represent a switch plane. In an IPU-POD64 there are 32 such planes and with 32 128-port 1U 100GbE switches it can be scaled to 8000 IPUs in a single switch hop. IPU-Fabric can connect clusters of IPU-PODs in scale-out from a few IPU-PODs to 1,000's of IPU-PODs and can scale to support a cluster of up to 64000 IPU's that can work as a singular AI compute entity or supporting 1000's of different workloads and tenants. The IPU-Fabric is fully compatible with 100Gb Ethernet using QSFP/OSFP connectors and standard switches. These can be used to connect IPU-Fabric clusters and to build larger systems and to fit in with existing datacentre technology[12].

The memory model for the IPU-Machine is also quite unique and in addition to In-IPU Memory each IPU-M2000 systems has DDR memory available to the four IPUs. This DDR memory is used differently from that found in CPUs or GPUs. Instead of a memory hierarchy that requires swapping data and code from host memory store to the accelerator's memory, the Poplar Graph Compiler creates the deterministic code-memory relationships in both the memory on the IPU tile and the DDR memory. In fact, the IPU-M2000 system can use this additional memory in stand-alone mode for inference processing without any attachment to a host server. And thanks to the BSP model compiling both computation and communication, the network communication overhead is kept to a minimum compared to traditional messaging or shared memory constructs commonly used for parallel processing.

Built-in fabrics are becoming a necessity for AI accelerators since model sizes are increasing dramatically, some containing billions of parameters. These large models must be distributed across hundreds or thousands of processors to solve problems in a reasonable time. Graphcore's hybrid model uses a proprietary IPU-Link fabric to communicate across the tiles in an IPU and adjacent rack IPUs, while tunnelling the IPU-Link protocol across standard 100GbE for rack-to-rack scale-out supporting larger configurations [12].

This disaggregated scaling model is the most important feature of IPU-M2000 based systems and, together with IPU-Fabric, enables a flexible disaggregation model, allowing the user to configure multiple accelerators on the fly without constraints by a predetermined scenario. It is also an important architectural element in the context of the hybrid type of system discussed in this article, where the HPC and AI part can be dynamically reconfigured based on the requirements and specifics of the code allocating HPC processors and accelerators in combination with the ML component based on the task requirements and code characteristics. It would be ideal to implement this in a multidimensional backbone that is efficiently supported by IPU-Fabric and an HPC network topology that would allow direct 1 to 1 communication with very low latency.

V. Programming heterogeneous systems

High performance scientific computing has traditionally focused on scaling and increasing the performance of a single large task. In fact, they were designed and optimized for the efficient execution of a few large-scale simulations, instead of the large number of smaller scale simulations necessary e.g., to train accurate ML models. However, even a new hybrid system as proposed in previous section of the article without a radical redesign of algorithms and computational methods faces increasingly serious limitations in the ability to scale single monolithic applications and achieve significant performance gains on large parallel machines. In response, HPC scientific applications are looking for new methods to reduce time to obtain scientific insight, such as the use of team - aggregated simulations [6], [13] and the integration of artificial intelligence and machine learning methodologies with traditional HPC applications. The coupling of ML methods and HPC simulations is not trivial and it involves a fundamental reconfiguration of classic HPC algorithms and the use of appropriate data sets. In particular, there are two design strategies:

- Sequential implementation of the simulation followed by AI / ML training and subsequent inference runs.
- Stream implementation of the AI / ML component that enables independent and concurrent running of simulation and AI / ML tasks, but with the possibility of data exchange by individual tasks at runtime. Tasks can still be interdependent, depending on how the simulations are selected using the AI / ML methods to run next.

The second strategy, although even more attractive and innovative, requires a complete rebuild of existing codes and the development of a completely new application, so most of the work and efforts of the researchers currently focuses on a strategy ensuring interaction between artificial intelligence and classical solvers for much faster analysis and reduced simulation time. Although using machine learning methods to speed up simulations can significantly improve the performance of scientific applications, there are many limitations that must be considered. ML systems require many examples (data samples) to build accurate surrogate models, and HPC systems are designed to perform as few as possible simultaneous instances of very complex tasks. This underlying tension between ML and HPC requires the problem of multiple simulations to be solved, unfortunately HPC is optimized for a few, meaning that the creation of HPC simulation datasets used to train ML models must be done with care. Standard HPC workflow tools may not be the most effective way to make the large simulation datasets required to train ML models. Also, batch scheduling systems are typically not designed to run thousands or millions of simulations. Parallel file systems can degrade performance when presented with a large number of simultaneous reads and writes that overload the metadata servers. Dynamically loaded shared objects can pose similar problems. Essentially, creating ML-ready HPC simulation datasets requires workflow technology that can efficiently coordinate asynchronous heterogeneous simulation tasks at scales well beyond the design operation of HPC systems [14].

Taking all this into account and being aware that this is the beginning of an arduous road, it should be emphasized that there are already several works and articles that can boast interesting results combining ML components with the classic HPC simulation.

Fundamentally these methods have been used to replace, accelerate, or enhance existing solvers via AI/ML solution. These methods are based on the fact that solvers compute a set of iterations to achieve the convergence state of the simulated phenomenon. These methods are based on the assumption of the convergence of ML models on the basis of several initial iterations generated by the classic solver. This way, intermediate iterations do not have to be computed to get the final result, and therefore the time needed to solve is significantly reduced. Fig. 6 shows the simplified scheme of replacing classical solver via ML enabled solution for a CFD workload [15].

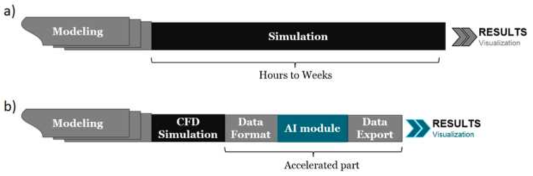


Fig. 6. Scheme of replacing classical solver via ML enabled solution

The presented approach includes the initial results computed by the CFD solver and the AI-accelerated part executed by the proposed AI module. The CFD solver produces results sequentially, iteration by iteration, where each iteration produces intermediate results of the simulation. All inter- mediate results wrap up into what is called the simulation results. The proposed method takes a set of initial iterations as an input, sends them to AI module, and generates the final iteration of the simulation. The AI module consists of three stages:

- data formatting and normalization,
- prediction with AI model (inference),
- data export

The advantage of this method is that it does not require to take into account a complex structure of the simulation, but focus on the data. Such an approach lowers the entry barrier for new adopters compared with other methods, such as a learning aware approach [15], which is based on the mathematical analysis of solver equations [15]. The AI-accelerated simulation is based on supervised learning, where a set of initial iterations is taken as an input and returns the last iteration. For simulating the selected phenomenon with conventional non-AI approach, it is required to execute 5000 iterations. At the same time, only the first iterations create the initial iterations that produce input data for the AI module. The accelerated part utilized a dedicated part of the system specialized for ML workloads, when CFD simulation portion uses HPC portion of the system. The proposed approach to accelerating CFD simulation allows shortens the simulation time almost ten times compared to using only a conventional CFD solver. The proposed AI / ML module uses 9.6% of the initial solver iterations and predicts a convergent state with an accuracy of 92.5% [15].

Undoubtedly, the 10-fold reduction in simulation time is impressive, and there are also studies in which this type of approach provides up to a 100-fold improvement in simulation related to protein folding. Over 100 times faster protein folding and 1.6 times more simulations per time unit, improving resource utilization compared to the classic HPC solver, even more interesting is that this type of approach is not used as an experiment, but is running on platforms and workloads in production [16].

In parallel to the hybrid approach that tries to combine HPC solutions with ML, there are several initiatives that use typical ML frameworks to solve classic HPC problems. This approach can be seen in the attempt to use TensorFlow to solve problems typical for HPC. TensorFlow was designed for creating ML applications, but it must be noted that it reduces the difficulty of programming accelerators in distributed environments, avoiding the low-level programming interfaces typical for HPC, such as CUDA, OpenCL and MPI. This concept allows the use of cloud systems with accelerators such as GPU, TPU, IPU, but also allows the use of typical HPC supercomputers. While TensorFlow was originally designed to solve machine learning problems, it can be generalized to solving a much wider range of numerical problems. Although TensorFlow is not a typical library for numerical computation, there are several examples of using TensorFlow to solve them. TensorFlow is a complete development environment with a high-level API that allows easy development and implementation of distributed algorithms without the need for in-depth knowledge of concepts such as CUDA and MPI. Although TensorFlow is a dynamically developed platform that can be used to create HPC applications on supercomputers with accelerators, it does not seem to be a good candidate to replace classic solvers due to the mismatch of the hardware architecture typical for ML (e.g. TPU or IPU) solutions, different from those for HPC [17].

Considering the proposed hybrid system architecture proposed in section IV where the ML component was based on the IPU-M2000, it should be noted that there are significant limitations in using the IPU for typical HPC tasks. Firstly, Poplar graphs are static, making it difficult to implement techniques such as dynamic grids and adaptive mesh refinement. Secondly, the graph compile time is very high compared to compilation of typical HPC kernels. It is important to emphasise that for small problems, graph compilation may take longer than executing the resulting programs. Thirdly, code developed for the IPU is not portable to other platforms. Fourth and most importantly, the IPU is limited to 32-bit precision, which is absolutely the biggest obstacle for some HPC scientific applications.

VI. Conclusions

The horizon of Exascale HPC projects is a set of challenges and opportunities. On the one hand, HPC methods and platforms are becoming general and necessary for scientific advances. On the other, classical HPC computations are reaching limits. The HPC community confidently expected that as long as improvements in hardware performance were promising, traditional simulation-based methods would continue to deliver improved performance. The ramifications of this belief are obvious, achieving productivity gains is becoming increasingly difficult, while at the same time requiring significant and unsustainable, investment in software and algorithmic reformulation. However, it is clear that traditional simulations may not represent the optimal approach for Exascale systems and for the next generations of supercomputers, which could consequently lead to complete stagnation.

The new discussed hybrid HPC and machine learning model enables a new approach to performance, scalability and execution time. In this new paradigm, a specialized ML system in conjunction with classical simulation replaces single large units, which requires both hierarchical and multitasking parallelism. The current research trend shows that hybrid systems, which are a combination of ML and HPC solutions, outperform simulation based approaches. The exact optimal point or intersection point is not trivial: it will be application specific, will depend on the complexity of the learned models, the amount of the data, and the effectiveness and cost of the simulation among others. However, the fundamental idea, that surrogate learned models will represent effective performance improvements over traditional simulations, is powerful and is an important generalization of the multi-scale, coarse-grained approaches used in many disciplines of science such as: computation fluid dynamics simulation, molecular science, climate change, materials simulation, and many others.

Presented details of surrogate system based on machine learning approach with high performance computing infrastructure that is widely available will have important implications for the cyberinfrastructure developed and deployed for the science of tomorrow. While a great deal of effort has gone into making the most of the HPC infrastructure, this article only describes a few fasteners and construction methods that seem easy to implement. From a future system architecture perspective ML methods have and will have an increasingly visible and important role in smarter computational systems. The main reason for the success of such techniques is that they offer simple, scalable and fairly general means to deal with high-dimensional, scientific datasets [14].

While new hybrid systems push the boundaries of effective simulation, this work demonstrates above all that a confluence of disaggregated ML components and traditional HPC technologies represents a promising path towards the realization of next generation integrated scientific computing.

References

[1] https://top500.org/statistics/list/

[2] Geoffrey Fox, James A. Glazier, JCS Kadupitiya, Vikram Jadhao, Minje Kim, Judy Qiu, James P. Sluka, Endre Somogyi, Madhav Marathe, Abhijin Adiga, Jiangzhuo Chen, Oliver Beckstein, and Shantenu Jha. "Learning Everywhere: Pervasive machine learning for effective High-Performance computation: Application background". Technical report, Indiana University, February 2019. http://dsc.soic.indiana.edu/ publications/Learning Everywhere.pdf.

[3] Geoffrey Fox, Shantenu Jha,"Understanding ML driven HPC: Applications and Infrastructure", Invited talk to "Visionary Track" at IEEE eScience 2019.

[4] Jeff Dean. "Machine learning for systems and systems for machine learning". In Presentation at 2017 Conference on Neural Information Processing Systems, 2017.

[5] Satoshi Matsuoka. "Post-K: A game changing supercomputer for convergence of HPC and big data" / AI. Multicore 2019, February 2019.

[6] Kadupitiya Kadupitige. "Intersection of HPC and Machine Learning". ENGR-E 687 IND STUDY INTEL SYS: FINAL REPORT

[7] https://docs.graphcore.ai/projects/ipu-overview/en/latest/about-ipu.html

[8] Leslie G. Valiant. 1990. "A bridging model for parallel computation". Commun. ACM 33, 8 (August 1990), 103-111.

[9] https://www.graphcore.ai/products/ipu

[10] Zhe Jia, Blake Tillman, Marco Maggioni, Daniele Paolo Scarpazza, "Dissecting the Graphcore IPU Architecture via Microbenchmarking", https://arxiv.org/abs/1912.03413

[11] Ion Stoica, Dawn Song, Raluca Ada Popa, David Patterson, Michael W. Mahoney, Randy Katz, Anthony D. Joseph, Michael Jordan, Joseph M. Hellerstein, Joseph Gonzalez, Ken Goldberg, Ali Ghodsi, David Culler, Pieter Abbeel. " A Berkeley View of Systems Challenges for AI". EECS Department. University of California, Berkeley. Technical Report No. UCB/EECS-2017-159. October 16, 201.7

[12] Karl Freund, Patrick Moorhead. "The Graphcore Second-Generation IPU". https://moorinsightsstrategy.com/research-paper-the-graphcore-second-generation-ipu/

[13] Thorsten Kurth, Sean Treichler, Joshua Romero, Mayur Mudigonda, Nathan Luehr, Everett Phillips, Ankur Mahesh, Michael Matheson, Jack Deslippe, Massimiliano Fatica, Prabhat, Michael Houston. "Exascale Deep Learning for Climate Analytics". Super Computing Conference November 11-16, 2018, Dallas, TX, USA

[14] J. Luc Peterson, Ben Bay, Joe Koning, Peter Robinson, Jessica Semler, Jeremy White, Rushil Anirudh, Kevin Athey, Peer-Timo Bremer, Francesco Di Natale, David Fox, Jim A. Gaffney, Sam A. Jacobs, Bhavya Kailkhura, Bogdan Kustowski, Steven Langer, Brian Spears, Jayaraman Thiagarajan, Brian Van Essen, Jae-Seung Yeom. "Enabling Machine Learning-Ready HPC Ensembles with Merlin". Lawrence Livermore National Laboratory, Livermore, California 94550, USA. https://arxiv.org/pdf/1912.02892.pdf

[15] Krzysztof Rojek, Roman Wyrzykowski, Pawel Gepner. "AI-Accelerated CFD Simulation Based on OpenFOAM and CPU/GPU Computing" International Conference on Computational Science -2021

[16] Alexander Brace, Hyungro Lee, Heng Ma, Anda Trifan, Matteo Turilli, Igor Yakushin, Todd Munson, Ian Foster, Shantenu Jha, Arvind Ramanathan. "Achieving 100X faster simulations of complex biological phenomena by coupling ML to HPC ensembles". https://arxiv.org/abs/2104.04797

[17] Steven W. D. Chien, Stefano Markidis, Vyacheslav Olshevsky, Yaroslav Bulatov, Erwin Laure, Jeffrey S. Vetter. "TensorFlow Doing HPC". https://arxiv.org/abs/1903.04364

DOI: 10.15439/2021F005
ISSN 2300-5963 ACSIS, Vol. 25

A Distributed Application Placement and Migration Management Techniques for Edge and Fog Computing Environments

Mohammad Goudarzi*, Marimuthu Palaniswami†, Rajkumar Buyya*,
*The Cloud Computing and Distributed Systems (CLOUDS) Laboratory
School of Computing and Information Systems
The University of Melbourne, Australia
Email: mgoudarzi@student.unimelb.edu.au, rbuyya@unimelb.edu.au
† The Department of Electrical and Electronic Engineering,
The University of Melbourne, Australia
Email: palani@unimelb.edu.au

***Abstract*—Fog/Edge computing model allows harnessing of resources in the proximity of the Internet of Things (IoT) devices to support various types of latency-sensitive IoT applications. However, due to the mobility of users and a wide range of IoT applications with different resource requirements, it is a challenging issue to satisfy these applications' requirements. The execution of IoT applications exclusively on one fog/edge server may not be always feasible due to limited resources, while the execution of IoT applications on different servers requires further collaboration and management among servers. Moreover, considering user mobility, some modules of each IoT application may require migration to other servers for execution, leading to service interruption and extra execution costs. In this article, we propose a new weighted cost model for hierarchical fog computing environments, in terms of the response time of IoT applications and energy consumption of IoT devices, to minimize the cost of running IoT applications and potential migrations. Besides, a distributed clustering technique is proposed to enable the collaborative execution of tasks, emitted from application modules, among servers. Also, we propose an application placement technique to minimize the overall cost of executing IoT applications on multiple servers in a distributed manner. Furthermore, a distributed migration management technique is proposed for the potential migration of applications' modules to other remote servers as the users move along their path. Besides, failure recovery methods are embedded in the clustering, application placement, and migration management techniques to recover from unpredicted failures. The performance results demonstrate that our technique significantly improves its counterparts in terms of placement deployment time, average execution cost of tasks, the total number of migrations, the total number of interrupted tasks, and cumulative migration cost.**

I. Introduction

THE NUMBER of latency-sensitive applications of Internet of Things (IoT) devices has been increasing due to recent advances in technologies so that many applications rely on remote resources for their execution. Due to latency-sensitive nature of these applications and the huge amount of data that they generate, traditional cloud computing cannot efficiently satisfy the requirements of IoT applications, and they experience high latency and energy consumption while communicating to cloud servers (CSs) [1], [2]. The fog/edge computing paradigm addresses these issues by providing an intermediate layer of distributed resources between IoT devices and CSs that can be accessed with lower latency [3], [4], [5]. However, the provided resources of fog/edge servers for IoT applications are limited and with less variety in comparison to the resources of CSs [6]. In our view, fog computing has a hierarchical and distributed structure that harnesses the resources of both CSs and Fog Servers (FSs) at different hierarchical fog levels, while lower-level FSs have fewer computing resources compared to higher-level FSs, but they are accessible with lower latency [1], [7], [8], [9], [10]. However, edge computing does not have this hierarchical structure and does not use resources of CSs [11] (although some works use these terms interchangeably).

Real-time IoT applications can be modeled as a set of lightweight and interdependent application modules in fog computing environments so that such application modules alongside their allocated resources form the data processing elements of various IoT applications [7], [1]. Considering different requirements of applications' modules, they can be placed on one FS, different FSs in the same hierarchical level, FSs in different hierarchical levels, and/or CSs for the execution [1], [12]. Besides, as the number of IoT applications increases, more requests are forwarded to FSs that may overload them. Hence, a dynamic application placement technique is required to efficiently place interdependent modules of IoT applications on remote servers while meeting their requirements.

Alongside the importance of suitable application placement techniques, there are yet several issues to be addressed. The coverage ranges of lower-level FSs are limited, and IoT users have different mobility patterns. Besides, interdependent modules of each IoT application may be deployed on several FSs. Hence, as the IoT user moves towards its destination, the application response time and IoT device energy consumption can be negatively affected [13]. Therefore, the migration of interdependent modules of each application among FSs, which incurs service interruption and additional cost, is an important and yet a challenging issue. Several migration techniques decide when, how, and where application modules can migrate when IoT users change their location in the fog/edge computing environments, such as [14], [15], [16], [17]. However, these techniques either focus on the migration of a single application module without considering other deployed modules [13] or consider an IoT application as a set of independent application modules. An IoT application may consist of several interdependent modules, and the migration technique should consider the configuration of all interdependent modules when an IoT user moves towards its destination. Hence, the migration of IoT applications, consisting of several interdependent modules, is an important challenge to be addressed, especially in hierarchical fog computing environments in which modules may be placed on different hierarchical levels.

Also, in fog computing, there are several studies that consider the application placement and migration management engines (i.e., decision engines) have a global view about topology and resources of all FSs and CSs [6], [18] while there are other studies that assume decision engines only have a local view about resources and topology of servers in their proximity [10], [9], [19]. In these latter techniques, the decision engines act in the distributed manner so that each FS that receives the application placement and/or migration request try to use the available resources in its proximity (which can be accessed with lower latency) to place/migrate the application modules as much as possible. However, if there are no available resources, the rest of the placement and migration will be handled by higher-level FSs in the hierarchy. Considering communication with higher-level FSs incurs higher latency compared to communication among FSs at the same hierarchical level, the clustering of FSs (if it is possible) at the same hierarchical level can provide sufficient resources (with less latency in comparison to higher-level FSs) to serve real-time IoT applications and reduce the amount of communication with higher-level FSs.

In this paper, we address these issues and propose efficient distributed application placement and migration management techniques to satisfy the requirements of real-time IoT applications while users move.

The main contributions of this paper are as follows.

- We propose a new weighted cost model based on IoT applications' response time and IoT devices' energy consumption for application placement and migration of IoT devices in hierarchical fog/edge computing environments to minimize cost of running real-time IoT applications.
- We put forward a dynamic and distributed clustering technique to form clusters of FSs at the same hierarchical levels so that such servers can collaboratively handle IoT application requirements with less execution cost.
- Considering the NP-Complete nature of application placement and migration problems in fog/edge computing environments, we propose a distributed application placement and migration management techniques to place/migrate modules of real-time applications on different levels of hierarchical architecture based on their requirements.
- We embed failure recovery methods in clustering, application placement, and migration management techniques to recover from unpredicted failures.

The rest of paper is organized as follows. Relevant works of application placement and migration management techniques in edge and fog computing environments are discussed in section II. The system model and problem formulations are presented in section III. Section IV presents our proposed distributed clustering, application placement, and migration management technique. We evaluate the performance of our technique and compare it with the state-of-the-art techniques in section V. Finally, section VI concludes the paper and draws future works.

II. Related Work

In this section, related works that address both application placement and mobility issues at the same time as their main challenges in the context of edge/fog computing are studied. These works are categorized into independent and dependent categories based on the dependency mode of their applications' granularity (e.g., modules). In the dependent category, constituent parts of IoT applications (i.e., modules) can be executed only when their predecessor modules complete their execution, while IoT applications that are modeled as a set of independent modules do not have this constraint.

A. *Edge Computing*

In the independent category, Wang et al. [20] formulated service migration as a distance-based Markov Decision Process (MDP), which considers the distance between an IoT user and service provider as its main parameter. Then, they proposed a numerical technique to minimize the migration cost of users. Wang et al. [21] and Yang et al [22] considered deterministic mobility conditions, in which the potential paths between source and destination are priori known, and proposed placement techniques, performed on the IoT device, to minimize the delay. Since paths and available edge devices are priori known, as the IoT user moves, the current in-contact edge device can send the required information to the next edge device. Ouyang et al. [17] proposed an edge-centric application placement and mobility management technique that are executed on the network operator and one-hop edge devices respectively. They proposed a distributed approximation scheme based on the best response update technique to optimize the mobile edge service performance. Liu et al. [23] proposed a mobility-aware offloading and migration technique to maximize the total revenue of IoT devices by reducing the probability of migration. Zhu et al. [24] proposed a mobility-aware application placement in vehicular scenarios with constraints on service latency and quality loss. In this technique, some of the vehicles generate tasks while other vehicles provide computing services as remote servers. Zhang et al. [25] proposed a deep reinforcement technique to minimize the delay of IoT tasks. Yu et al. [26] proposed a technique to minimize the delay of tasks while satisfying the energy consumption of a single IoT user moving among edge servers.

In the dependent category, Sun et al. [27] and Qi et al. [28] proposed a mobility-aware application placement technique in which placement decision engines run on IoT devices. The authors of [27] considered a single IoT device and proposed an IoT-centric energy-aware mobility management technique to minimize the application delay while authors of [28] proposed an edge-centric and knowledge-driven online learning method to adapt to the environment changes as vehicles move.

B. Fog Computing

In the independent category, Wang et al.[16] proposed a solution to place a single service instance of each IoT user on a remote server when multiple IoT users exist in the system. They proposed both offline and online approximation algorithms, performed on the cloud, to find the optimal and near-optimal solutions respectively. Wang et al. [29] and Wang et al. [13] proposed edge-centric application placement and mobility management technique when multiple IoT users with a single module exist in the system. The main goal of [29] is Maximizing IoT users' gain through offloading and reducing the number of migrations, while the main goal of authors of [13] is minimizing the service delay.

In the dependent category, Shekhar et al. [6] and Bittencourt et al. [19] proposed mobility-aware application placement techniques for IoT application, consisting of multiple inter-dependent modules while considering prior mobility information. The authors in [6] proposed a cloud-centric technique, called URMILA, in which the centralized controller makes the placement decision for all IoT applications to satisfy their latency requirements. Besides, whenever the decision is made, even in case the user leaves the range of its immediate server, there is no migration algorithm to migrate modules to new servers, which incurs a higher cost for the users. The authors in [19] proposed an edge-centric solution based on the edgeward-placement technique [10] for placement of IoT applications while considering their targeted destination. In this proposal, however, the potential of clustering is not considered. So, whenever the immediate server cannot serve the application modules, the modules are forwarded to the next hierarchical layer for possible placement and migration.

C. A Qualitative Comparison

Key elements of related works are identified and presented in Table I and compared with ours in terms of the main category, IoT application, architectural, and placement and mobility management engines' properties. The IoT application properties identify and compare dependency mode (either independent or dependent) of IoT applications, modules' number (either single or multiple modules per application), and heterogeneity (whether the specification of modules is same (i.e., homogeneous) or different (i.e., heterogeneous)). Architectural properties contain the number of IoT devices (either single or multiple), whether hierarchical fog architecture is considered or not, and clustering technique (whether a clustering technique is applied on edge/fog servers or not). Placement and mobility management engines contain positions of placement, mobility management engines, failure recovery capability, and the decision parameters used in each proposal.

Our work proposes an edge-centric application placement and mobility management technique for an environment consisting of multiple IoT devices with heterogeneous applications (consisting of several dependent modules with heterogeneous requirements) and multiple remote servers (either CSs or FSs) deployed in a hierarchical architecture. Considering the potential of the clustering of FSs in the hierarchical fog computing environment, we propose a weighted cost model of response time and energy consumption for the application placement and migration techniques. The proposed weighted cost model considers the dependency among modules of IoT applications which plays an important role in application placement and migration management. Second, we put forward a distributed and dynamic clustering technique by which FSs of the same hierarchical level can form a cluster and collaboratively provide faster and more efficient service for IoT applications. This latter is because the communication overhead between FSs of the same hierarchical level is usually less than communication with higher-level FSs [1]. Although resources of each lower-level FS is less than each higher-level FS, aggregated resources of lower-level FSs, obtained through clustering, can be used to manage IoT applications modules in lower-level FSs with less response time and energy consumption. Third, we propose a distributed application placement and migration techniques for hierarchical fog computing environments to minimize the weighted cost of running real-time IoT applications. Finally, due to the highly dynamic nature of such systems, there is a high chance of failures in the system, for which we propose light-weight failure recovery methods in the clustering, application placement, and migration management techniques.

III. System Model and Problem Formulation

We consider a system consisting of N mobile IoT users (so that each user has one IoT device), F heterogeneous FSs distributed in the proximity of IoT users, and a centralized cloud. FSs follow a hierarchical topology, in which lower-level FSs can be accessed with lower latency while providing fewer resources in comparison to higher-level FSs that provide more resources but can be accessed with higher latency [1], [9]. Besides, we assume that each IoT device is connected to one FS in the lowest hierarchical level, so that this FS is responsible for the application placement and mobility management of that IoT device. The set of all available servers is represented as $\mathcal{S}$ with $|\mathcal{S}| = M$ and $M > F$. The 2-tuple $(h, i) \in \mathcal{S}$ $(0 \leq h, 1 \leq i)$ represents one server, in which h represents the hierarchical level of the server and i denotes the server's index at that hierarchical level. If we assume there are L hierarchical fog layers, $(L+1, 1)$ demonstrates the centralized cloud data-center placed at the top-most level. Moreover, the $(0, n)$ denotes the nth IoT device. Fig. 1 represents a view of our system model and how IoT devices move among different FSs. Also, it shows the in-cluster communications (in case clustering is applied) and communications between FSs at different hierarchical levels in this environment.

Table I: A qualitative comparison of related works with ours

Techniques	Category	Application Properties			Architectural Properties			Placement and Mobility Management Engines					
		Dependency	Module Number	Heterogeneity	Number of IoT Devices	Hierarchical Fog Architecture	Clustering Technique	Placement Engine Position	Mobility Management Engine Position	Failure Recovery	Decision Parameters		
											Time	Energy	Weighted
[20]	Edge Computing	Independent	Single	Heterogeneous	Single	×	×	Edge Centric	Edge Centric	×	✓	×	×
[21]			Multiple	Heterogeneous	Multiple	×	×	IoT Device Centric	Edge Centric	×	✓	×	×
[17]			Single	Heterogeneous	Multiple	×	×	Edge Centric	Edge Centric	×	✓	×	×
[23]			Single	Heterogeneous	Multiple	×	×	Edge Centric	Edge Centric	×	✓	✓	✓
[22]			Multiple	Heterogeneous	Multiple	×	×	IoT Device Centric	Edge Centric	×	✓	×	×
[24]			Multiple	Heterogeneous	Single	×	×	Edge Centric	Edge Centric	×	✓	×	×
[25]			Single	Homogeneous	Single	×	×	Edge Centric	Edge Centric	×	✓	×	×
[26]			Multiple	Heterogeneous	Single	×	×	Edge Centric	Edge Centric	×	✓	✓	×
[27]		Dependent	Multiple	Heterogeneous	Single	×	×	IoT Device Centric	IoT Device Centric	×	✓	✓	×
[28]			Multiple	Heterogeneous	Multiple	×	×	IoT Device Centric	Edge Centric	×	✓	×	×
[16]	Fog Computing	Independent	Single	Heterogeneous	Multiple	×	×	Cloud Centric	Cloud/Edge Centric	×	✓	×	×
[29]			Single	Heterogeneous	Multiple	×	×	Edge Centric	Edge Centric	×	✓	✓	✓
[13]			Single	Heterogeneous	Multiple	×	×	Edge Centric	Edge Centric	×	✓	×	×
[6]		Dependent	Multiple	Homogeneous	Single	×	×	Cloud Centric	Cloud Centric	×	✓	×	×
[19]			Multiple	Heterogeneous	Multiple	✓	×	Edge Centric	Edge Centric	×	✓	×	×
Proposed Solution			Multiple	Heterogeneous	Multiple	✓	✓	Edge Centric	Edge Centric	✓	✓	✓	✓

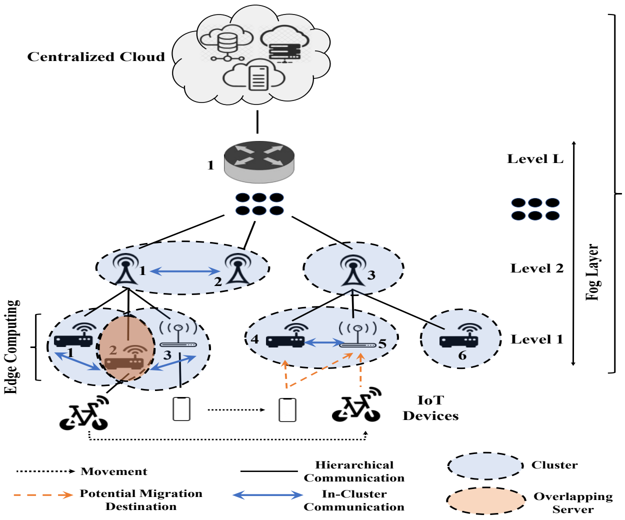


Figure 1: A view of our system model

Each FS can form a cluster either by other nearby FSs at the same hierarchical level or by itself. Moreover, each FS in lth hierarchical level may belong to different clusters in that hierarchical layer. The cluster member (CM) list of each FS is defined as $List_{cl}(h, i)$, which is empty if the FS (h, i) does not have any CMs. Besides, for each FS, we define a children list, $List_{ch}(h, i)$, containing server specification of immediate lower-level FSs, to which it has direct hierarchical communication links. The sole parent server of each FS is defined as $par(h, i) = (h', i')$ which refers to the immediate higher-level FS. We assume that in-cluster communications are faster than hierarchical communications [1]. Hence, clustering FSs, while incurs additional cost due to running clustering algorithm, can improve the quality of service for IoT users. Moreover, each FS has a list, called $\Omega(h, i)$, containing server specification of itself, its children, and all FSs belonging to the Ω of its children. To illustrate, considering Fig. 1, the $\Omega(2, 1) = \{(2, 1), (1, 1), (1, 2), (1, 3)\}$ and $List_{ch}(2, 1) = \{(1, 1), (1, 2), (1, 3)\}$, and $\Omega(2, 2) = \{(2, 2)\}$ and $List_{ch}(2, 2) = \{\}$. If we assume the maximum number of fog layers is three (i.e., $L = 3$) in this example, then $\Omega(3, 1) = \{(3, 1), (2, 1), (2, 2), (2, 3), (1, 1), (1, 2), (1, 3), (1, 4)$ $, (1, 5), (1, 6)\}$, and the $List_{ch}(3, 1) = \{(2, 1), (2, 2), (2, 3)\}$.

We consider that FSs and CSs use container technology to run IoT applications' modules [13], [30]. So, we assume that FSs have access to images of all containers ($Cnts$) while such $Cnts$ may be active if they are running on the server or inactive (i.e., the container images are accessible, but the containers are not running) otherwise [13]. Moreover, for each container, according to the application module that it serves, an amount of ram size at the runtime is assigned to keep the state $Cnt^{ram}_{v_{n,j}}$ [31]. Table II summarizes the parameters used in this paper and their respective definitions.

A. Application Model

We consider real-time IoT applications working based on the Sense-Process-Actuate model, in which sensors transmit tasks periodically according to their sample rate [1], [10]. The emitted sensors' tasks should be forwarded to different modules of the IoT applications for processing based on dependency model among constituent modules. When each module receives tasks from predecessor modules as input, it processes tasks and produces respective tasks as its output to be forwarded to next modules [1], [32]. Finally, results

Table II: Parameters and respective definitions

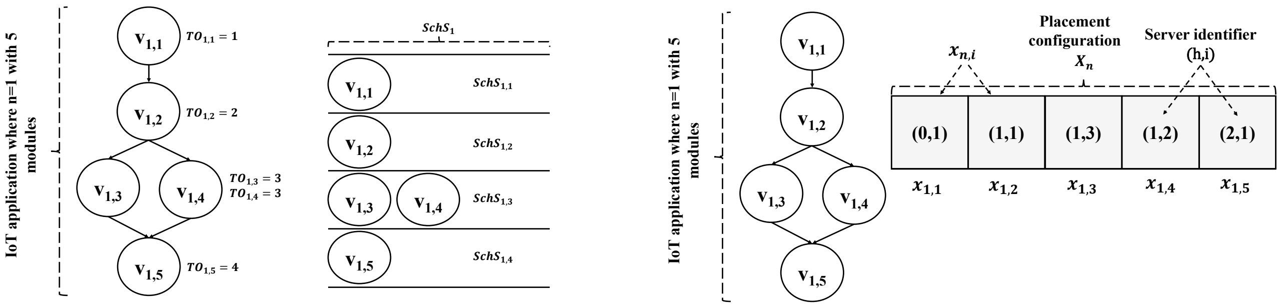
Parameter	Definition	Parameter	Definition
CSs	Cloud Servers	FSs, FS	Fog Servers, Fog Server
$CNTs$, CNT	Containers, Container	N	Number of mobile IoT devices
F	Number of heterogeneous fog servers (FSs)	$\mathcal{S}$	The set of all available servers
M	Number of available servers	(h, i)	The 2-tuple showing one server in which h represents the hierarchical level of the server and i denotes the server's index at that hierarchical level
$List_{ch}(h, i)$	The list containing server specification of children for the server (h, i)	$par(h, i)$	The sole parent of the server (h, i) in the hierarchical system
$\Omega(h, i)$	The set containing server specification of server (h, i), its children, and all FSs belonging to the Ω of its children	CM	Cluster Member
$List_{cl}(h, i)$	The list containing server specification of cluster members for the server (h, i)	G_n	Directed Acyclic Graph (DAG) of the nth IoT application
$\mathcal{V}_n$	The set of modules belonging to the nth IoT application	$\mathcal{E}_n$	The set of data flows between modules belonging to the nth IoT application
$v_{n,i}$, $v_{n,j}$	The ith and/or jth module belonging to the nth IoT application	$e_{n,i,j}$	The data flow from module $v_{n,i}$ to module $v_{n,j}$ of the nth IoT device
$\mathcal{P}(v_{n,j})$	The set of predecessor modules of the module $v_{n,j}$	$TO_{n,i} = t$	The topological order of ith module of the nth IoT application is equalt to t
$SchS_n$	The schedule set of the nth IoT application consisting of subsets of modules with the the same TO value t	$SchSn, t$	A subset of $SchS_n$ showing modules with the same TO value t (i.e., modules that can be executed in parallel)
$e^{ins}_{n,i,j}$	The amount of instructions in terms of Million Instruction that the module $v_{n,j}$ receives from $v_{n,i}$ for processing	$e^{dsize}_{n,i,j}$	The size of data that the module $v_{n,i}$ generates as an output to be sent to module $v_{n,j}$
$v^{mtd}_{n,i}$	The maximum tolerable delay for the module $v_{n,i}$	X_n	The placement configuration of the nth IoT application
$x_{n,i}$	The placement configuration for each module $v_{n,i}$ of the nth IoT application in the X_n	$\Psi(X_n, t)$	The weighted cost of modules in the tth schedule while considering the placement configuration X_n.
$\lvert SchS_n \rvert$	The number of schedules for the nth IoT application	$T_{x_{n,j}}$	The overall delay of each module (i.e., $v_{n,j}$) based on its assigned server
$Cnts_{(h,i)}$	The number of instantiated $Cnts$ on the server (h, i)	$Cap_{(h,i)}$	The maximum capacity of server (h, i) to instantiate $Cnts$.
$\Gamma(X_n, t)$	The weighted cost of modules in the tth schedule while considering the placement configuration X_n	$\Theta(X_n, t)$	The energy consumption of modules in the tth schedule while considering the placement configuration X_n
$T^{lat}_{x_{n,j}}$	The inter-nodal latency between the servers on which module $v_{n,j}$ and its predecessors $\mathcal{P}(v_{n,j})$ are placed	$T^{exe}_{x_{n,j}}$	The computing execution time of tasks, emitted from the $v_{n,i}$ to be executed on the $v_{n,j}$
$T^{tra}_{x_{n,j}}$	The transmission time between between the module $v_{n,j}$ and its predecessors $\mathcal{P}(v_{n,j})$	$cpu(x_{n,j})$	The computing power of the assigned server (in terms of MIPS) for the module $v_{n,j}$
γ^{tra}	The transmission time between source and destination servers	B_{up}, B_{down}, $B_{cluster}$	The bandwidth of the one server to the parent server, to the child server, and to its CMs, respectively
$NST_i(H), NSE_i(H)$	They define the next intermediate server to reach the destination server	$chRule$	It identifies whether any children of the current server has a route to the destination server or not
$chRule$	It identifies whether any CMs of the current server has a route to the destination server or not	$\Upsilon((\Omega(h, i)), (h', i'))$	It shows whether $\Omega(h, i)$ contains (h', i') or not (i.e., meaning that there is one hierarchical path from (h, i) to the (h', i'))
γ^{lat}	The inter-nodal latency between source and destination servers	$lat(up)$, $lat(down)$, $lat(cluster)$	The inter-nodal latency of one server to the parent server, to the child server, and to its CMs, respectively
$\Psi^{mig}((X_n, X'_n, ts)$	The weighted migration cost of nth IoT application from the current configuration X_n to the new configuration X'_n	$\gamma^{mig}(x_{n,i}, x'_{n,i})$	The migration cost of one module from current configuration $x_{n,i}$ to the new configuration $x'_{n,i}$
$\gamma^{lat}_{mig}((h, i), (h', i'))$	The migration latency between current and new servers	$dsize^{mig}$	The size of dump data and states that should be transferred between current and new servers
$e^{ins,r}_{n,i,j}$	The amount of remaining instructions of task $e^{ins,r}_{n,i,j}$ to be executed on the new server after migration	$E(x_{n,j})$	The overall energy consumption of each module (i.e., $v_{n,j}$) based on its assigned server
$E^{exe}_{x_{n,j}}$	The computing energy consumption of tasks, emitted from the $v_{n,i}$to be executed on the $v_{n,j}$	$E^{lat}_{x_{n,j}}$	The energy consumption incurred due to inter-nodal latency between the servers on which module $v_{n,j}$and its predecessors $\mathcal{P}(v_{n,j})$ are placed
$E^{tra}_{x_{n,j}}$	The transmission energy consumption between between the module $v_{n,j}$ and its predecessors $\mathcal{P}(v_{n,j})$	P_{cpu}, P_i, P_t	The CPU power of the IoT device, the idle power of IoT device, and transmission power of the IoT device
ϑ^{tra}	The transmission energy consumption between source and destination servers	ϑ^{lat}	The energy consumption incurred due to inter-nodal latency between servers
$\Gamma^{mig}((X_n, X'_n), t)$	The migration time of nth IoT application from the current configuration X_n to the new configuration X'_n considering schedule t	$\Theta^{mig}((X_n, X'_n), t)$	The migration energy consumption of nth IoT application from the current configuration X_n to the new configuration X'_n considering schedule t

will be forwarded to the actuator as the last module. In this work, we assume that both sensor and actuator modules of IoT applications reside in IoT devices [19].

Real-time IoT application belonging to the nth IoT device is represented as a Directed Acyclic Graph (DAG) of its modules $G_n = (\mathcal{V}_n, \mathcal{E}_n), \forall n \in \{1, 2, \cdots, N\}$, where $\mathcal{V}_n = \{v_{n,i} | 1 \leq i \leq |\mathcal{V}_n|\}$ denotes the set of modules belonging to the nth IoT device, and $\mathcal{E}_n = \{e_{n,i,j} | v_{n,i}, v_{n,j} \in \mathcal{V}_n,\ v_{n,i} \in \mathcal{P}(v_{n,j}),\ i \neq j\}$ shows the set of data flows between modules. Since IoT applications are modeled as DAGs, each module $v_{n,j}$ cannot be executed unless all its predecessor modules, denoted as $\mathcal{P}(v_{n,j})$, finish their execution. To illustrate, $e_{1,1,2}$ represents that execution of module $v_{1,2}$ depends on the execution of the module $v_{1,1}$. Moreover, we define a Topological Order value t for each module i of the nth IoT application as $TO_{n,i} = t$. We define a schedule set for the nth IoT application, called $SchS_n$, consisting of modules with the same TO value t as its subsets. The $SchS_{n,t}$ specify modules with the same TO value t (i.e., modules that can be executed in parallel). In addition, the set of successor modules of module $v_{n,j}$ is defined as $Succ(v_{n,j})$. Fig 2a shows an IoT application, the TO value for each module, and the schedule set $SchS_n$ based on the TO values of its modules. Besides, We define the output of each module $v_{n,i}$ is a task consisting of two values to be forwarded to next modules based on data flows of the IoT application. The first value is the amount of instructions in terms of Million Instruction (MI) that the module $v_{n,j}$ receives from $v_{n,i}$ for processing, shown as $e^{ins}_{n,i,j}$, and the second value is the size of data $e^{dsize}_{n,i,j}$ the module $v_{n,i}$ generates as its output to be forwarded to module $v_{n,j}$ [10].

B. Problem Formulation

The placement configuration of the application belonging to the nth IoT application is shown as X_n. Also, $x_{n,i} = (h, i)$ denotes the placement configuration for each module $v_{n,i}$ of



(a) An IoT application and its corresponding schedules

(b) A candidate server configuration for the IoT application

Figure 2: An example of IoT application, its schedules and a candidate server configuration

the nth IoT application in the X_n based on the specification of the server. To illustrate, $x_{n,i} = (1,3)$ shows that the ith module of nth IoT device is assigned to a server in the first hierarchical level where the server index is 3. Moreover, if the ith module of the nth IoT device is assigned to run locally on itself, $x_{n,i} = (0,n)$. Fig 2b presents a sample DAG of an IoT application and a candidate placement configuration.

1) Placement weighted cost model: The goal of application placement is to find a suitable configuration for modules of each real-time IoT application to minimize the weighted cost $\Psi(X_n, t)$ of running applications in terms of the response time of tasks and energy consumption of IoT devices:

$$\min_{w_1, w_2 \in [0,1]} \sum_{t=1}^{|SchS_n|} \Psi(X_n, t), \quad \forall n \in \{1, 2, \cdots, N\} \tag{1}$$

where

$$\Psi(X_n, t) = w_1 \times \Gamma(X_n, t) + w_2 \times \Theta(X_n, t) \tag{2}$$

$$s.t. \quad C1: \; Size(x_{n,j}) = 1, \; \forall x_{n,j} \in X_n \;, \quad n \in \{1, 2, \cdots, N\}, 1 \leq i \leq |\mathcal{V}_n| \tag{3}$$

$$C2: \; Cnts(h,i) \leq Cap(h,i), \; \forall \; (h,i) \in \mathcal{S} \tag{4}$$

$$C3: \; \Psi(x_{n,i}, t) \leq \Psi(x_{n,j}, t), \; \forall v_{n,i} \in \mathcal{P}(v_{n,j}) \tag{5}$$

where $|SchS_n|$ represents the number of schedules, and $\Gamma(X_n, t)$ and $\Theta(X_n, t)$ show the response time model and energy consumption model, respectively, of modules in the tth schedule while considering the placement configuration X_n. Moreover, w_1 and w_2 are control parameters to tune the weighted cost model according to user requirements. We assume the number of available servers M is more than or equal to the maximum number of modules in the tth schedule for parallel execution (i.e., $|SchS_{n,t}| \leq M$). We suppose that each module of an IoT application can be exactly assigned to one Cnt of one remote server. $C1$ indicates that each module i of the nth IoT application can only be assigned to one server at a time, and hence the size of $x_{n,j}$ is equal to 1 [2], [33]. $C2$ denotes that the number of instantiated $Cnts$ on the server (h,i) is less or equal to the maximum capacity of that server $Cap(h,i)$. Besides, $C3$ guarantees that the predecessor modules of $v_{n,j}$ (i.e., $\mathcal{P}(v_{n,j})$) are executed before the execution of module $v_{n,j}$ [33].

*a) **Response time model:*** The goal of this model is to find the best possible configuration of servers for each IoT application so that the overall response time for each IoT application becomes minimized. In order to only consider response time model as the main objective, the control parameters of weighted cost model (Eq. 2) can be set to $w_1 = 1$ and $w_2 = 0$.

$$\Gamma(X_n, t) = \begin{cases} T(x_{n,j}), & \text{if } |SchS_{n,t}| = 1 \quad \text{(a)} \\ max(T(x_{n,j})), & \text{otherwise} \\ \forall x_{n,j} \in X_n | v_{n,j} \in SchS_{n,t} & \quad \text{(b)} \end{cases} \tag{6}$$

The Eq. 6.a represents the condition in which the number of modules in the tth schedule is one (i.e, $|SchS_{n,t}| = 1$), and hence, the time of that schedule is equal to the time of that module based on its assigned server $T(x_{n,j})$. Besides, the Eq. 6.b refers to the condition in which the number of modules in the tth schedule is more than one (i.e., several modules can be executed in parallel). In this latter case, the time of the tth schedule is equal to the maximum time of all modules that can be executed in parallel.

The overall delay of each module (i.e., $v_{n,j}$) based on its candidate configuration (i.e., $x_{n,j}$) is defined as the sum of inter-nodal latency between servers ($T^{lat}_{x_{n,j}}$), the computing time per module ($T^{exe}_{x_{n,j}}$), and the data transmission time between $v_{n,j}$ and all of its predecessor modules ($T^{tra}_{x_{n,j}}$). It is formulated as:

$$T(x_{n,j}) = T^{exe}_{x_{n,j}} + T^{lat}_{x_{n,j}} + T^{tra}_{x_{n,j}} \tag{7}$$

The computing execution time of module $v_{n,j}$ depends on tasks emitted from its predecessors (i.e., $\mathcal{P}(v_{n,j})$) for processing by $v_{n,j}$. The computing time of $v_{n,j}$ is estimated as:

$$T^{exe}_{x_{n,j}} = \sum \frac{e^{ins}_{n,i,j}}{cpu(x_{n,j})}, \quad \forall e_{n,i,j} \in \mathcal{E}_n | v_{n,i} \in \mathcal{P}(v_{n,j}), \tag{8}$$

where $cpu(x_{n,j})$ demonstrates the computing power of the assigned server (in terms of Million Instruction per Second (MIPS)) for the module $v_{n,j}$. Moreover, the $e^{ins}_{n,i,j}$ shows the amount of instructions in terms of MI that the module $v_{n,j}$ receives from $v_{n,i}$ for the processing.

The transmission time between module $v_{n,j}$ and its predecessors $\mathcal{P}(v_{n,j})$ of the application belonging to the nth IoT device is calculated as:

$$\begin{aligned} T^{tra}_{x_{n,j}} = \max(\gamma^{tra}(e^{dsize}_{n,i,j},(h,i),(h',i'))),\\ \forall e_{n,i,j} \in \mathcal{E}_n | v_{n,i} \in \mathcal{P}(v_{n,j}),\\ x_{n,i}=(h,i), x_{n,j}=(h',i') \end{aligned} \tag{9}$$

Due to the hierarchical nature of fog computing, the transmission time of one task (γ^{tra}) between each pair of dependent modules $v_{n,i}$ and $v_{n,j}$ is recursively obtained based on visited servers between source and destination. The (h,i) and (h',i') show server specifications of source and destination servers on which modules $v_{n,i}$ and $v_{n,j}$ are assigned, respectively. By visiting each intermediate server between source and destination servers, the value of source server (h,i) is updated while the value of destination server remains unchanged. To reduce the length of equations, we consider $(e^{dsize}_{n,i,j},(h,i),(h',i')) = H$.

$$\gamma^{tra}(H) = \begin{cases} \frac{e^{dsize}_{n,i,j}}{B_{up}} + \gamma^{tra}(H'), & NST_i(H) = NST_1|NST_4|NST_6 \\ \frac{e^{dsize}_{n,i,j}}{B_{down}} + \gamma^{tra}(H'), & NST_i(H) = NST_2 \\ \frac{e^{dsize}_{n,i,j}}{B_{cluster}} + \gamma^{tra}(H'), & NST_i(H) = NST_3|NST_5 \\ 0, & NST_i(H) = NST_7 \end{cases} \tag{10}$$

where B_{up}, B_{down}, and $B_{cluster}$ refer to the bandwidth of current server to parent server, to child server, and to cluster server, respectively. Besides, H' is defined as what follows:

$$H' = (e^{dsize}_{n,i,j},(h^{''},i^{''}),(h',i')) \tag{11}$$

$$(h^{''},i^{''}) = NST_i(H) \tag{12}$$

The Eq. 11 shows the data size and destination server of H' is exactly the same as H, and the only difference is the specification of the source server $(h^{''},i^{''})$ which is obtained from the output of $NST_i(H)$ (i.e., $(h^{''},i^{''}) = NST_i(H)$. The $NST_i(H)$ defines the next intermediate server to reach the destination server for each edge $e_{n,i,j}$.

$$NST_i(H) = \begin{cases} Par(h,i), & \text{if } h < h' & i=1 \\ chRule, & \begin{array}{l}\text{if } h > h' \\ \& \ chRule \neq \varnothing\end{array} & i=2 \\ clRule, & \begin{array}{l}\text{if } h \oplus h' = 0 \\ \& \ i \oplus i' \neq 0 \\ \& \ clRule \neq \varnothing\end{array} & i=3 \\ Par(h,i), & \begin{array}{l}\text{if } h \oplus h' = 0 \\ \& \ i \oplus i' \neq 0 \\ \& \ clRule = \varnothing\end{array} & i=4 \\ clRule, & \begin{array}{l}\text{if } h > h', \\ \& \ clRule \neq \varnothing\end{array} & i=5 \\ Par(h,i), & \begin{array}{l}\text{if } h > h' \\ \& \ chRule = \varnothing \\ \& \ clRule = \varnothing\end{array} & i=6 \\ (0,0), & \begin{array}{l}\text{if } h \oplus h' = 0 \\ \& \ i \oplus i' = 0\end{array} & i=7 \end{cases} \tag{13}$$

$$\begin{aligned} chRule = \text{if } \exists(h^{''},i^{''}) \in List_{ch}(h,i)| \\ \Upsilon((\Omega(h^{''},i^{''}),(h',i')) = 1, \text{return } (h^{''},i^{''}), \\ \text{else return } \varnothing \end{aligned} \tag{14}$$

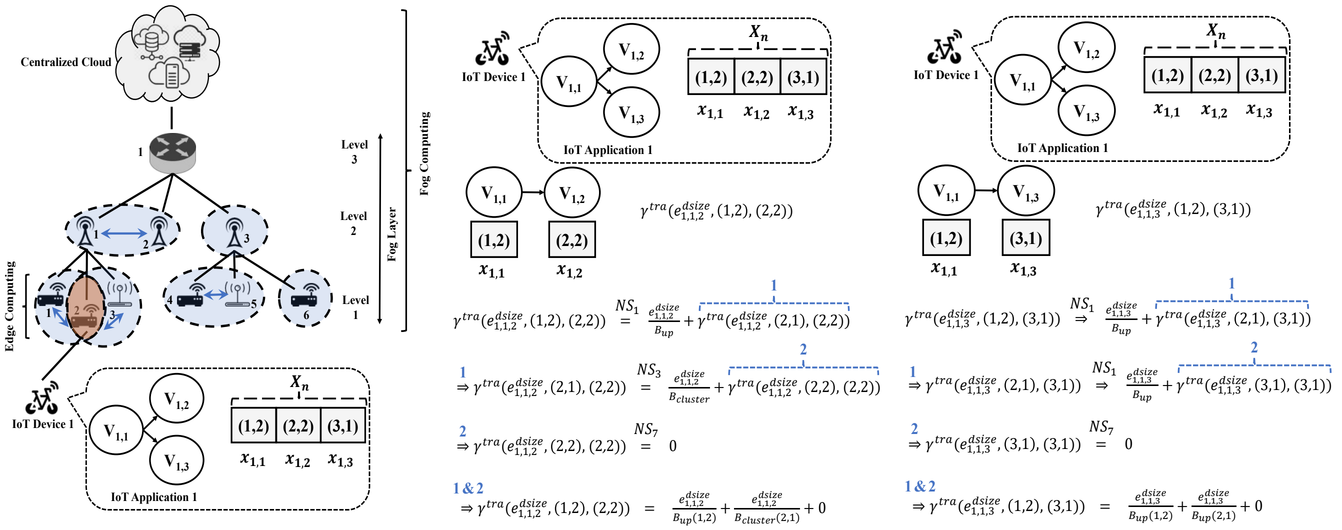
$$\begin{aligned} clRule = \text{if } \exists(h^{''},i^{''}) \in List_{cl}(h,i)| \\ \Upsilon((\Omega(h^{''},i^{''}),(h',i')) = 1, \text{return } (h^{''},i^{''}), \\ \text{else return } \varnothing \end{aligned} \tag{15}$$

The $\Upsilon((\Omega(h^{''},i^{''})),(h',i'))$ is equal to 1 if $\Omega(h^{''},i^{''})$ contains (h',i') (i.e., meaning that there is one hierarchical path from $(h^{''},i^{''})$ to the (h',i')) and is equal to 0 if (h',i') does not exist. Moreover, the $\oplus$ is XOR binary operation. The chRule (Eq. 14) says that if the server (h,i) has a children $(h^{''},i^{''})$ in its $List_{ch}$ which has a hierarchical path to the destination server $(h^{'},i^{'})$, the specification of this server $(h^{''},i^{''})$ should be returned. The clRule (Eq. 15) presents that if the server (h,i) has a CM $(h^{''},i^{''})$ in its $List_{cl}(h,i)$ which has a hierarchical path to the destination server $(h^{'},i^{'})$, the specification of this server $(h^{''},i^{''})$ should be returned. Based on the aforementioned rules, $NST(H)$ finds the next server to which the data should be sent and calculates the transmission cost. The NST_1 of Eq. 13 states that if the hierarchical level of the current server is less than destination server, the $Par(h,i)$ should be checked in the next step. The NST_2 represents the case that the hierarchical level of the current server is higher than the destination server and the current server has a child through which the destination server can be reached. The NST_3 states the condition that the current and destination servers are in the same hierarchical level, and one of the CMs has a route to the destination server. The NST_4 indicates that if the current and the destination servers are in the same level, and there is no route to destination using CMs, the parent should be checked in the next step. The NST_5 states that if the level of the current server is higher than the destination server, and a CM has a path to the destination server, the cluster server should be selected in the next step. The NST_6 states that if the level of current server is higher than the destination server, and there exists no route from children nor from CMs, the parent server should be traversed. Finally, the NST_7 is the ending condition for this recursive process and states that if the current and destination server is same, the cost is zero. Fig 3 represents an example of obtaining transmission time between source and destination servers.

Inter-nodal latency $T^{lat}_{x_{n,j}}$ between servers on which module $v_{n,j}$ and its predecessors $\mathcal{P}(v_{n,j})$ are placed is calculated as:

$$\begin{aligned} \Gamma^{lat}_{X_{n,j}} = \max(\gamma^{lat}((h,i),(h',i'))),\\ \forall e_{n,i,j} \in \mathcal{E}_n | v_{n,i} \in \mathcal{P}(v_{n,j}),\\ x_{n,i}=(h,i), x_{n,j}=(h',i') \end{aligned} \tag{16}$$

where γ^{lat} shows the inter-nodal latency between source and destination servers (i.e., (h,i) and (h',i') respectively) on



(a) An IoT application and its candidate configuration

(b) Transmission time for the data flow $e_{1,1,2}$

(c) Transmission time for the data flow $e_{1,1,3}$

Figure 3: A example of calculating transmission time based on a candidate configuration

which $v_{n,i}$ and $v_{n,j}$ are placed. It is calculated similar to the transmission time. To reduce the equation size, we consider $((h,i),(h',i'))=A$.

$$\gamma^{lat}(A)=\begin{cases} lat_{up}+\gamma^{lat}(A'), & NST_i(A)=NST_1|NST_4|NST_6\\ lat_{down}+\gamma^{lat}(A'), & NST_i(A)=NST_2\\ lat_{cluster}+\gamma^{lat}(A'), & NST_i(A)=NST_3|NST_5\\ 0, & NST_i(A)=NST_7 \end{cases} \tag{17}$$

where lat_{up}, lat_{down}, and $lat_{cluster}$ correspond to up-link, down-link, and cluster-link inter-nodal latency respectively, and depends on the hierarchical level of servers. Besides, A' is defined as what follows:

$$A'=((h'',i''),(h',i')) \tag{18}$$

The Eq. 18 shows the destination server (i.e., (h',i')) of A' is exactly the same as A, and the only difference is the specification of the source server (h'',i'') which is obtained from the output of $NST(A)$. The $NST(A)$ performs exactly the same as $NST(H)$ (i.e., Eq. 13) to find the next intermediate server, and all equation from Eq. 13 to Eq. 15 are valid here.

*b) **Energy consumption model**:* The goal of this model is to find a suitable placement configuration of application modules to minimize the energy consumption of the nth IoT device. To only consider energy consumption model as the main objective, the control parameters of weighted cost model (Eq. 2) can be set to $w_1=0$ and $w_2=1$.

$$\Theta(X_n,t)=\begin{cases} E(x_{n,j}), & \text{if } |SchS_{n,t}|=1 \quad \text{(a)}\\ max(E(x_{n,j})), & \text{otherwise} \\ \forall x_{n,j}\in X_n|v_{n,j}\in SchS_{n,t} & \quad \text{(b)} \end{cases} \tag{19}$$

where $|SchS_n|$ shows the number of schedules, and $\Theta(X_n,t)$ represents the energy consumption of modules in the tth schedule while considering the placement configuration X_n.

The overall energy consumption of each module (i.e., $v_{n,j}$) based on its candidate configuration (i.e., $x_{n,j}$) is defined as the sum of energy consumed for inter-nodal latency between servers ($E^{lat}_{x_{n,j}}$), the computing of each module ($E^{exe}_{x_{n,j}}$), and the data transmission between $v_{n,j}$ and all of its predecessor modules ($E^{tra}_{x_{n,j}}$). It is formulated as:

$$E(x_{n,j})=E^{exe}_{x_{n,j}}+E^{lat}_{x_{n,j}}+E^{tra}_{x_{n,j}} \tag{20}$$

The computing energy consumption for module $v_{n,j}$ depends on its assigned server and can be derived from:

$$E^{exe}_{x_{n,j}}=\begin{cases} T^{exe}_{x_{n,j}}\times P_{cpu}, & \text{if } x_{n,j}=(h,i)\ \&\ h=0\\ T^{idle}_{x_{n,j}}\times P_i, & \text{if } x_{n,j}=(h,i)\ \&\ h\neq 0 \end{cases} \tag{21}$$

Because only the energy consumption of IoT devices is considered in this work, whenever application modules run on remote servers, the energy consumption of IoT device is equal to the idle time $T^{idle}_{x_{n,j}}$ multiplied to the power consumption of IoT device in its idle mode P_i. Besides, P_{cpu} is the CPU power of the IoT device on which the module $v_{n,j}$ runs.

The energy consumption for data transmission between the module $v_{n,j}$ and its predecessors $\mathcal{P}(v_{n,j})$ of the application belonging to the nth IoT device is calculated as follows:

$$\begin{aligned} E^{tra}_{x_{n,j}} &= \max(\vartheta^{tra}(e^{dsize}_{n,i,j},(h,i),(h',i'))), \\ &\forall e_{n,i,j}\in\mathcal{E}_n|v_{n,i}\in\mathcal{P}(v_{n,j}), \\ &x_{n,i}=(h,i),x_{n,j}=(h',i') \end{aligned} \tag{22}$$

where, to reduce the length of equations, we consider $H=(e^{dsize}_{n,i,j},(h,i),(h',i'))$. Similar to response time model, (h,i) and (h',i') show the specifications of source and destination servers on which modules $v_{n,i}$ and $v_{n,j}$ runs, respectively. The transmission energy consumption between each pair of dependent modules $(\vartheta^{tra}(H))$ is calculated as follows:

$$\vartheta^{tra}(H) = \begin{cases} (\frac{e^{dsize}_{n,i,j}}{B_{up}} \times P_t) + (\gamma^{tra}(H') \times P_i), & NSE_i(H) = NSE_1 \\ (\frac{e^{dsize}_{n,i,j}}{B_{down}} \times P_t) + (\gamma^{tra}(H') \times P_i), & NSE_i(H) = NSE_2 \\ \gamma^{tra}(H') \times P_i, & NSE_i(H) = NSE_3 \end{cases} \tag{23}$$

where P_t presents the transmission power of the IoT device, and the NSE_i shows transmission configuration based on H.

$$NSE_i(H) = \begin{cases} H' = (e^{dsize}_{n,i,j}, Par(h,i), (h',i')), & \text{if } h < h' \text{ \& } i = 1 \\ & \quad h = 0, \\ H' = (e^{dsize}_{n,i,j}, (h,i), Par(h',i')), & \text{if } h > h' \text{ \& } i = 2 \\ & \quad h' = 0, \\ H' = H, & \text{otherwise, } i = 3 \end{cases} \tag{24}$$

NSE_1 states the data flow is starting from an IoT device as the source server to remote servers as destination. Hence, the respective transmission energy consumption is equal to the required time to send the data to the parent server of IoT device multiplied by P_t, plus the IoT device's idle time (in which the data is transmitted from parent server to the destination) multiplied by P_i. Moreover, NSE_2 represents the invocation starting from remote servers as the source to the IoT device as the destination. It is important to note that the transmission power of IoT device P_t is active only if one of the modules is assigned to the IoT device and another module run on the remote servers, because we only consider the energy consumption from the IoT device's perspective. In other conditions, the transmission energy consumption is equal to the transmission time γ^{tra} (obtained from Eq. 10), in which the IoT device is in idle mode, multiplied by P_i (NSE_3).

The inter-nodal energy consumption $E^{lat}_{x_{n,j}}$ between servers on which module $v_{n,j}$ and its predecessors $\mathcal{P}(v_{n,j})$ are placed is calculated as:

$$\begin{aligned} E^{lat}_{X_{n,j}} &= \max(\vartheta^{lat}((h,i),(h',i'))), \\ &\forall e_{n,i,j} \in \mathcal{E}_n | v_{n,i} \in \mathcal{P}(v_{n,j}), \\ &x_{n,i} = (h,i), x_{n,j} = (h',i') \end{aligned} \tag{25}$$

where ϑ^{lat} shows the energy consumption incurred due to inter-nodal delay between source and destination servers on which $v_{n,i}$ and $v_{n,j}$ are placed. This latter is calculated similar to transmission energy consumption based on the $NSE_i(A)$ [33], [2]. To reduce the equation size, $((h,i),(h',i')) = A$.

$$\vartheta^{lat}(A) = \gamma^{lat}(A) \times P_i \tag{26}$$

where the $\gamma^{lat}(A)$ is obtained from Eq. 17.

2) Migration weighted cost model: We assume that the migration of modules belonging to the nth IoT device from current servers to new servers only happens due to the mobility of IoT devices. We consider pre-copy memory migration in which the current servers still running while transferring pre-dump to the new servers [13], [31]. The goal of migration cost model is to minimize the the downtime plus required cost of executing remaining instructions on the new servers. The migration weighted cost model is defined as:

$$\min_{w_1,w_2 \in [0,1]} \Psi^{mig}((X_n, X'_n), t), \quad \forall t \in |SchS_n|, \quad \forall n \in \{1, 2, \cdots, N\} \tag{27}$$

where

$$\Psi^{mig}((X_n, X'_n), t) = w_1 \times \Gamma^{mig}((X_n, X'_n), t) + w_2 \times \Theta^{mig}((X_n, X'_n), t) \tag{28}$$

$$s.t. \qquad C1: \sum_{t=1}^{|SchS_n|} \Psi(X'_n, t) \leq \sum_{t=1}^{|SchS_n|} \Psi(X_n, t) + \epsilon \tag{29}$$

where $\Gamma^{mig}((X_n, X'_n), t)$ and $\Theta^{mig}((X_n, X'_n), t)$ represent the additional time and energy consumption incurred by the migration of modules of tth schedule in the downtime (when the service is interrupted). The C1 states the service cost for tasks emitted from modules of nth IoT device in the new configuration X'_n should be less or roughly the same while considering the previous configuration X_n. The ϵ shows an acceptable additional service cost in the migration. Moreover, constraints C1, C2, and C3 from Eq. 1 are valid here as well.

*a) **Migration time model**:* The migration time is considered as the execution time required to finish remaining instructions on the new servers plus the downtime. This latter includes the time for suspending the $Cnts$ in current servers, transmission of the dump and states, and $Cnts$' resuming time on the new servers. Since, in the downtime, a specific amount of dump data and states should also be transferred between servers ($dsize^{mig}$), the migration latency $\gamma^{lat}_{mig}((h,i),(h',i'))$ and migration transmission time between current and new servers $\gamma^{tra}_{mig}(dsize^{mig},(h,i),(h',i'))$ to transfer this data are also important [31]. Besides, the $Cnts$' stopping time plus its resuming time are considered as a constant I^{mig}. The migration time is defined as:

$$\begin{aligned} \Gamma^{mig}((X_n, X'_n), t) &= Max(\gamma^{mig}(x_{n,i}, x'_{n,i})), \\ &\forall x_{n,i} \in X_n, \forall x'_{n,i} \in X'_n | v_{n,i} \in SchS_{n,t}, \\ &x_{n,i} = (h,i), x^{'}_{n,i} = (h',i') \end{aligned} \tag{30}$$

where

$$\begin{aligned} \gamma^{mig}(x_{n,i}, x'_{n,i}) &= \gamma^{lat}_{mig}((h,i),(h',i')) + I^{mig} \\ &+ \gamma^{tra}_{mig}(dsize^{mig},(h,i),(h',i')) + \frac{e^{ins,r}_{n,i,j}}{cpu(x'_{n,i})} \end{aligned} \tag{31}$$

where $\gamma^{mig}(x_{n,i}, x'_{n,i})$ represents the migration cost of module $v_{n,i}$ from its current server $x_{n,i}$ to its new server $x'_{n,i}$. The γ^{tra}_{mig} and γ^{lat}_{mig} are calculated based on 10 and 17, respectively. Also, $\frac{e^{ins,r}_{n,i,j}}{cpu(x'_{n,i})}$ shows the execution time of remaining instructions of task $e^{ins,r}_{n,i,j}$ on the new server (h',i').

*b) **Migration energy consumption model**:* The additional energy consumption of IoT device, incurred by the migration, depends on the execution of remaining instructions and the downtime.

$$\begin{aligned} \Theta^{mig}((X_n, X'_n), t) &= Max(\vartheta^{mig}(x_{n,i}, x'_{n,i})), \\ &\forall x_{n,i} \in X_n, \forall x'_{n,i} \in X'_n | v_{n,i} \in SchS_{n,t}, \end{aligned} \tag{32}$$

$$x_{n,i} = (h, i), x^{'}_{n,i} = (h', i')$$

where

$$\vartheta^{mig}(x_{n,i}, x'_{n,i}) = \vartheta^{lat}_{mig}((h, i), (h', i')) + I^{mig} \\ + \vartheta^{tra}_{mig}(dsize^{mig}, (h, i), (h', i')) + \vartheta^{exe}_{mig}(x'_{n,i}) \tag{33}$$

where $\vartheta^{mig}(x_{n,i}, x'_{n,i})$ represents the amount of energy consumed by the IoT device in the migration of each module of application from its current server $x_{n,i}$ to its new server $x^{'}_{n,i}$. The ϑ^{tra}_{mig} and ϑ^{lat}_{mig} represent the energy consumption incurred due to the transmission and migration latency between current and new servers. They are calculated based on 23 and 26, respectively. Also, the $\vartheta^{exe}_{mig}(x'_{n,j})$ shows the energy consumption required for the execution of remaining instructions of task $e^{ins,r}_{n,i,j}$ on the new server (h', i').

$$\vartheta^{exe}_{mig}(x'_{n,i}) = \begin{cases} \frac{e^{ins,r}_{n,i,j}}{cpu(x'_{n,i})} \times P_{cpu}, & \text{if } x'_{n,i} = (h', i') \text{ \& } h' = 0 \\ \frac{e^{ins,r}_{n,i,j}}{cpu(x'_{n,i})} \times P_{i}, & \text{if } x'_{n,i} = (h', i') \text{ \& } h' \neq 0 \end{cases} \tag{34}$$

C. Optimal Decision Time Complexity

We assume M servers exist in the hierarchical fog/edge computing environment and the maximum number of modules in each IoT application is K. Each module of an IoT application can be assigned to one of the M candidate servers at a time. Hence, for an IoT application with K modules, the Time Complexity (TC) of finding the global optimal solution for the application placement and the migration is $O(M^K)$. This cost is prohibitively high and prevents us from obtaining the global optimal solution in real-time [34]. Hence, we propose distributed algorithms to find an acceptable solution in a polynomial time for application placement and migration techniques in hierarchical fog computing environments.

IV. Proposed Technique

In this section, we present a fog server architecture to support distributed application placement, migration management, and clustering (as depicted in Fig. 4) by extending the fog server architecture proposed in [1]. Each FS in [1] is composed of three main components: controller, computational, and communication. We extend this architecture to support clustering and mobility management of IoT users in a distributed manner.

In our FS architecture, the Controller Component monitors and manages the Communication and Computational Components. It consists of three decision engine blocks and several meta-data blocks to store important information. The *Clustering Engine* is responsible for forming a distributed cluster with its in-range FSs and updating CMs' information in the *Cluster Info* and *Routing Info* meta-data. The *Application Placement Engine* is responsible for placement of IoT applications' modules to minimize the overall cost of running real-time IoT applications. It checks *Cluster Info*, *Resource Info*, and *Routing Info* meta-data for making placement decision, and updates the *Placement Info* and *Resource Info* meta-data blocks to store the configuration of application modules and available resources in this FS, respectively. The *Migration Management Engine* of each FS controls migration process of applications' modules when IoT users move. This module considers all meta-data blocks including the current mobility information of the users (i.,e *Mobility Info*), and decides the migration destination of application modules. Based on its decision, *Placement Info* and *Resource Info* will be updated to store last changes in the configuration of application modules.

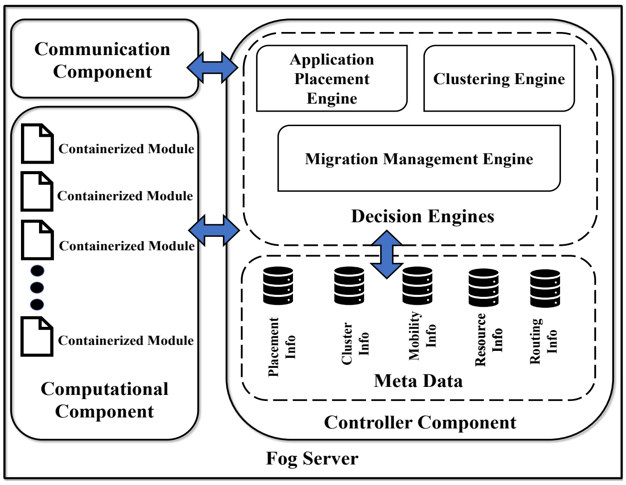


Figure 4: A view of fog server architecture

The Computational Component provides resources for the execution of application modules that are assigned to this FS based on the container technology. Besides, the Communication Component is responsible for network functionalities such as routing and packet forwarding, just to mention a few [1].

A. Dynamic Distributed Clustering

Since FSs usually have fewer resources in comparison to CSs, one FS may not be able to provide service for all modules of one application. Moreover, in some scenarios, several IoT devices are connected to the same FS, and hence, the FS may not be able to serve all application modules of different IoT devices due to its limited resources. Thus, other modules of one application should be placed on either CSs or higher-level FSs for the execution. However, in a hierarchical fog computing environment, in which the potential clustering of FSs is considered, application modules can be placed or migrated to other FSs in the same cluster. It can reduce the placement and migration cost of application modules.

We consider that FSs belonging to the same hierarchical layer can form a cluster by any in-range FSs at the same hierarchical level and swiftly communicate together using the Constrained Application Protocol (CoAP), Simple Network Management Protocol (SNMP), and so forth. Therefore, the communication delay within a cluster is lower than communication using up-link and down-link [1]. Besides, in a reliable IoT-enabled system, it is expected that the fog infrastructure providers have applied efficient networking techniques to ensure steady communication among the FSs through less variable inter-nodal latency [1]. Algorithm 1 provides an overview of the dynamic distributed clustering technique.

When an FS joins the network, it receives and stores *CandidParent* control messages from FSs residing in the immediate upper layer. The new FS finds coordinates of its position and estimates the average latency to all candidate parents. It selects the FS with the minimum distance as its parent and sends an acknowledgment to it using *ParentSelection* method. Moreover, the new FS broadcasts a *FogJoining* control message, containing its position and coverage range, to its one-hop neighbors (lines 2-7). FSs receiving this message send back a *replyNewFog* control message with their list of active and inactive $Cnts$, positions' specifications, and their coverage range to the new FS. Besides, they update their CM list $List_{cl}$ with specifications of this new FS (lines 8-14). As the new FS receives *replyNewFog* message, it builds its CM list $List_{cl}$ with specifications of FSs residing in the same hierarchical layer. Alongside storing lists of active and inactive $Cnts$ of its CMs, positions, and their coverage range (lines 15-21). This distributed mechanism helps FSs to dynamically update their CM lists when a new FS joins the network.

We consider that each FS can leave the network in normal conditions (e.g., when the low-level FS is switched off by its user) or due to a failure (such as hardware or software failures). Before an FS leaves the network in normal conditions either permanently or temporarily, we assume that all of its assigned tasks should be finished. Hence, it only needs to send *StartFogLeaving* control message to its CMs to update the $List_{cl}$ of themselves, to its parent server, and to its children to find a new parent (lines 22-25). All FSs that receive *FogLeaving* control message remove all information related to this FS from their entries. Also, the children of the leaving FS that receive this control message call the *ParentSelection* method to update their parent (lines 26-32). In case of a fatal error, in which the leaving FS cannot send a control message to the parent, CMs, and children, its immediate parent runs the *StartFogFailureRecovery* and sends *FogFailureRecovery* control message to its children list $List_{ch}$ so that they can remove entries related to the failed FS (lines 33-39). It is important to note that this latter process takes more time in comparison to the *FogLeaving* process in normal conditions due to the higher latency of uplink and downlink communications. Besides, if any FS children loose their connection to their parent, they can run the *ParentSelection* method to choose a new parent.

In addition, each FS sends the latest information about its $List_{ch}$ to its parent FS if any changes happen. This helps higher-level FSs update their Ω.

B. Application Placement

Due to the time consuming nature of finding the optimal solution (Section III-C) for the application placement problem, a Distributed application placement technique (DAPT) is proposed to find a well-suited solution in a distributed manner (Algorithm 2). The DAPT starts whenever an application placement request arrives, and the serving FS tries to place application modules on appropriate servers so that real-time tasks, emitted from modules, can be processed with the minimum cost. Considering the weighted cost (Eq. 1), DAPT attempts to place modules of IoT applications in one/several FSs on the lowest-possible layer while considering the potential of clustering. However, if available resources in that/those FSs are not sufficient, it considers upper layer FSs or/and CSs to place the rest of modules. In this way, DAPT reduces the search space of Eq. 1 for each FS by only considering itself, its parent FS, and its CMs, and aims at reducing the overall weighted cost. Moreover, a distributed failure recovery method is embedded in DAPT to recover from possible failures.

Algorithm 1: Dynamic distributed clustering

```
   Input   : $RCM$: Received Control Message
1  switch $RCM$ do
2      case $CandidParent$ do
3          ParentSelection()
4          message.add(getPosition(),coverRange)
5          message.type($FogJoining$)
6          Broadcast(message)
7      end
8      case $FogJoining$ do
9          message.add(getPosition(),coverRange)
10         message.add(getActiveCnts(),getInactiveCnts())
11         message.type($ReplyNewFog$)
12         send($RCM$.getSourceAddr(), message)
13         $List_{cl}$.update($RCM$.getData())
14     end
15     case $ReplyNewFog$ do
16         $List_{cl}$.update($RCM$.getData())
17         $MapActiveCnt_{cl}$.put($RCM$.getSourceAddr(),
18         message.getListActiveCnts())
19         $MapInActiveCnt_{cl}$.put($RCM$.getSourceAddr(),
20         message.getListInActiveCnts())
21     end
22     case $StartFogLeaving$ do
23         message.type($FogLeaving$)
24         Broadcast(message)
25     end
26     case $FogLeaving$ do
27         $List_{cl}$.remove($RCM$.getSourceAddr())
28         $List_{ch}$.remove($RCM$.getSourceAddr())
29         if $RCM$.getSourceAddr() $==$ this.Parent) then
30             ParentSelection()
31         end
32     end
33     case $StartFogFailureRecovery$ do
34         for $i = 1$ to $List_{ch}$.size() do
35             message.type($FogFailureRecovery$)
36             message.setFailedFog(failedFog.getAddr())
37             send($List_{ch}$.get(i).getSourceAddr(),message)
38         end
39     end
40     case $FogFailureRecovery$ do
41         $List_{cl}$.remove($RCM$.getFailedFogAddr())
42     end
43 end
```

The immediate FS that receives the placement request from an IoT device is considered as the application placement controller ($controller$) for that IoT device. If the controller is performing the placement of a set of modules or a parent FS receives placement request from its children and the failure recovery mode is not active (lines 3-28), the *ClusterCheck* method returns the list of CMs and their available resources (line 4). Then, the list of ready servers S_R containing parent FS, current FS, and available CMs is created (line 5). This list contains all servers that current FS considers for the placement of modules in that hierarchical layer. Next, the *FindOrder* method checks either topological order of modules (TO_n) are available or not. If it is not available, it considers the DAG $\mathcal{G}_n$ of nth IoT application, and using the Breadth-First-Search (BFS) Algorithm finds topological order of all modules, and

Algorithm 2: An overview of DAPT

```
   Input  : $\mathcal{G}_n$: The DAG of $n$th IoT device, $U_{\mathcal{G}_n}$: A subset of unassigned
            modules from $\mathcal{G}_n$, $X_n$: The configuration of assigned modules,
            $controller_{ID}$: ID of the placement controller
   Output : $X_n$
1  $s_{ID}$: this.ID
2  $List_{cl}$: this.getClusterMembers()
3  if (controller(n) || ReqFromChild) & !DAPTFailureRecovery(n) then
4      $List^{A}_{cl}$=ClusterCheck($List_{cl}$)
5      $S_R$=ReadyServers($List^{A}_{cl}$,this.parent,$s_{ID}$)
6      $SchS_n$=FindOrder($\mathcal{G}_n$)
7      $U_{(\mathcal{G}_n)}$=Sort($U_{(\mathcal{G}_n)}$, $SchS_n$)
8      if $S_R - Par(s_ID) \neq \varnothing$ then
9          for $i = 1$ to $U_{\mathcal{G}_n}$.size() do
10             v=$U_{(\mathcal{G}_n),i}$
11             $ID_{min}$=FindMinCost($S_R$,$\mathcal{G}_n$,$X_n$,$v$)
12             if $ID_{min}$ == $s_{ID}$ then
13                 $res_v$=CalService(v)
14                 if this.$Cnts$.contains(v) & then
15                     ScaleCnts(v,$res_v$)
16                 else
17                     StartCnt(v)
18                 end
19                 UpdateConfig($X_n$,v,$s_{ID}$)
20             end
21             else
22                 $ReqList$.update($v$,$ID_{min}$)
23             end
24         end
25         PlaceReqToServers($ReqList$,$\mathcal{G}_n$,$X_n$,$S_R$,$TO_n$,$SchS_n$)
26     else
27         PlacePar($\mathcal{G}_n$,$U_{\mathcal{G}_n}$,$X_n$,$TO_n$,$SchS_n$)
28     end
29 else if !controller(n) & !DAPTFailureRecovery(n) then
30     for $i = 1$ to $U_{\mathcal{G}_n}$.size() do
31         v=$U_{(\mathcal{G}_n),i}$
32         $res_v$=CalService(v)
33         if this.$Cnts$.contains(v) & $res_v$ $\leq$ this.Resource then
34             ScaleCnts(v,$res_v$)
35             UpdateConfig($X_n$,v,$s_{ID}$)
36             NotifyController(v, $s_{ID}$,$controller_{ID}$)
37         else
38             if $res_v$ $\leq$ this.Resource then
39                 StartCnt(v)
40                 UpdateConfig($X_n$,v,$s_{ID}$)
41                 NotifyController(v, $s_{ID}$,$controller_{ID}$)
42             else
43                 SendDAPTFailureRecovery(n,v,$controller_{ID}$,$s_{ID}$)
44             end
45         end
46     end
47 else
48     DAPTFailureRecovery(n,v,$S_R$,$X_n$)
49 end
```

creates $SchS_n$ (line 6). This latter helps to identify modules that do not have any dependency and can be executed in parallel. Then, $Sort$ method defines priority value for modules that can be executed in parallel (i.e., modules with the same topological order) based on non-increasing order of their rank value (line 7). The rank of each module is defined as:

$$Rank(v_{n,j}) = \begin{cases} C^{exe}_{n,j} + \max(C^{tra}_{n,j,z} + Rank(v_{n,z})) & \text{if } v_{n,j} \neq exit \\ \forall v_{n,z} \in Succ(v_{n,j}), & \\ & \\ C^{exe}_{n,j}, & \text{if } v_{n,j} = exit \end{cases} \tag{35}$$

where $C^{exe}_{n,j}$ shows the average weighted execution cost of module $v_{n,j}$, and $C^{tra}_{n,j,z}$ depicts the transmission cost of module $v_{n,j}$ and $v_{n,z}$, which are calculated as:

$$C^{exe}_{n,j} = w_1 \times \widetilde{T^{exe}_{x_{n,j}}(S_R)} + w_2 \times \widetilde{E^{exe}_{x_{n,j}}(S_R)} \tag{36}$$

$$C^{tra}_{n,j,z} = w_1 \times \widetilde{\gamma^{tra}_{n,j,z}(S_R)} + w_2 \times \widetilde{\vartheta^{tra}_{n,j,z}(S_R)} \tag{37}$$

where $\widetilde{T^{exe}_{x_{n,j}}(S_R)}$ and $\widetilde{E^{exe}_{x_{n,j}}(S_R)}$ show the average execution time and energy consumption of each module considering available servers in the S_R. The execution time $T^{exe}_{x_{n,j}}$ and energy consumption $E^{exe}_{x_{n,j}}$ of each module per server are obtained from Eq. 8 and Eq. 21 respectively. Besides, $\widetilde{\gamma^{tra}_{n,j,z}(S_R)}$ and $\widetilde{\vartheta^{tra}_{n,j,z}(S_R)}$ shows the average transmission time and energy consumption between modules $v_{n,j}$ and $v_{n,z}$ considering available servers in the S_R. The transmission time $\gamma^{tra}_{n,j,z}$ and transmission energy consumption $\vartheta^{tra}_{n,j,z}$ between each pair of servers in the S_R can be obtained from Eq. 10 and Eq. 23, respectively. Moreover, w_1 and w_2 are control parameters to tune the weighted cost. The rank is calculated recursively by traversing the DAG of application, starting from the exit module. The $Sort$ method can find the critical path of the DAG and gives higher priority to the modules that incur higher execution cost among modules that can be executed in parallel. Hence, the probability of placement of these modules on lower-level FSs increases. This latter is important since the resources of lower-level FSs are limited compared to higher-level FSs, but they can be accessed with less communication cost. Hence, if modules are more communication and latency-sensitive, they can be placed on lower-level FSs with higher priority while if they are computation-intensive modules, that cannot be efficiently executed on the lower-level FSs, they can be forwarded to higher-level FSs with higher priority. If S_R contains any candidate server except its parent, for each module v of $U_{\mathcal{G}_n}$, the *FindMinCost* receives the S_R, $\mathcal{G}_n$, and configuration X_n, as its input and finds the minimum cost for the execution of the module v based on current solution configuration X_n (i.e., based on the assigned servers' configuration to the predecessors of this module). Although in fog computing environments, a large number of FSs are deployed as candidate servers, the DAPT only considers FSs in the S_R, to which the serving FS can communicate with the lowest possible transmission and inter-nodal cost. Moreover, we assume that FSs do not have a global view of all FSs in the environment. Therefore, the search space in each hierarchical layer is reduced while the suitable candidate servers for real-time and latency-sensitive IoT applications are kept. After prioritizing modules, the execution cost of each module based on the available servers in S_R is calculated using *FindMinCost* method. This method checks the available resources required to run or scale the $Cnts$ to run these modules on available servers. Then, among the servers that meet these requirements, it returns the ID of the selected FS, ID_{min}, that can execute module v while minimizing the overall application cost using Eq.1 (line 11). If the current FS is selected, and it has active Cnt, the *ScaleCnt* method scales the resources so that it can serve this module (line 15). If there is no active Cnt in this FS, it should run a new Cnt, which incurs a Cnt startup cost (line 17). The candidate solution configuration X_n is updated accordingly so that the new configuration can be considered for the placement of the rest of the modules (line 19). If the selected FS is among the CMs or parent FS, the module v and its corresponding assigned server are stored in the request list $ReqList$ (line 22) so that it can be forwarded to their destination using the *PlaceReqToServers* (line 25). This method sends modules to assigned serves along with the topological order of this IoT application TO_n, schedules $SchS_n$, and current solution configuration X_n. Finally, in a case that the S_R is empty, meaning that the current controller does not have any resources and also it does not have any candidate servers with sufficient resources, it sends all modules to the parent FS so that the placement can be started in the higher hierarchical levels by means of the *PlacePar* method (line 27). If the parent FS receives the placement request from its children, it checks the possibility of placement of received modules on its S_R. The background reason is if one FS receives some modules for placement from its children FSs, it means that those modules are either more computation-intensive rather than latency/communication-intensive, or the children FSs did not have sufficient resources for these modules. However, if one FS receives a placement request from its CMs, it starts the deployment of modules on the condition that the available resources meet the modules' requirements.

If serving FS is not the controller FS and the failure recovery mode is not active (i.e, the placement request is forwarded to CMs), it iterates over the received modules (i.e., $U_{\mathcal{G}_n}$) and calculates the required amount of resources for each module *CalService(v)*. If it has enough resources, it starts the module, and using *NotifyController* method sends an acknowledgment for the controller FS. However, if due to any problem this FS cannot place this module, it runs *SendDAPTFailureRecovery* method, which sends a failure message to the controller FS so that the controller can make a new decision (lines 29-47).

If failure recovery mode is active, it means that one or several servers cannot properly execute assigned modules. Hence, the DAPT algorithm calls *DAPTFailureRecovery* method. This method receives failed modules of nth IoT application and finds corresponding FSs from the solution configuration X_n. If it has several candidate servers in S_R, it removes specification of the failed FS from S_R. Then, it iterates over the rest of available servers to finds FSs for these modules that minimize the execution cost. However, if the current FS only has its parent sever in the S_R, *DAPTFailureRecovery* sends a control message to activate *DAPTFailureRecovery* method of the parent FS. (line 48). It helps to check the possibility of placement of these modules in higher hierarchical layers.

C. Migration Management Technique (MMT)

As the user of nth IoT device is moving away from its current low-level FS (i.e., its controller FS) to a new low-level FS, the current controller FS should initiate the migration process to find a new controller FS, and migrate the current data and states of running $Cnts$ to new FSs. We suppose IoT devices can detect distributed low-level FSs (eg., using beacons, GPS, etc) and update their list of sensed FSs $List^n_{SFog}$ periodically. Whenever the controller FS realizes that the IoT device n is about to leave (e.g., through the received signal to noise ratio), it receives $List^n_{SFog}$ from the IoT device and initiates the migration process. The goals of the migration management technique (MMT) is to 1) find a new controller FS with the maximum sojourn time for the IoT device and 2) find a set of substitute servers for processing of IoT application's modules while minimizing the migration cost (Eq. 27). The Algorithm 3 shows an overview of the distributed migration process.

Whenever a controller FS realizes the nth IoT device is about to leave its coverage range, it initiates *MigrationInitiate* to find a new controller FS for the IoT device. The current controller FS receives the list of sensed low-level FSs $List^n_{SFog}$ from nth IoT device and removes its s_{ID} from this list so that it cannot be selected as a new controller FS (line 4). The mobility information of each user *mobInfo(n)* contains its average speed and its direction. Moreover, in the clustering technique, each FS learns the position and coverage ranges of its CMs. Considering the aforementioned values, the controller FS can estimate the sojourn time of this IoT device for each CM. The *MobilityAnalyzer* method (line 5) receives *mobInfo(n)* and $List^n_{SFog}$ and checks whether the $List^n_{SFog}$ contains any CMs of the current FS controller. Moreover, it finds specifications of other FSs belonging to $List^n_{SFog}$ through its CMs, if possible. The *MobilityAnalyzer* then creates two separate lists for reachable FSs ($List_{reach}$) and unreachable FSs ($List_{unreach}$) from $List^n_{SFog}$. The former one contains any FSs of $List^n_{SFog}$ which are among CMs of the current controller FS or those that can be accessed through its CMs, while the latter one refers to FSs to which the controller FS does not have access either directly or through its CMs. The *MobilityAnalyzer* method gives higher priority to FSs of $List_{reach}$ because the required information for the new controller to start its procedures can be more efficiently transferred to these FSs compared to those FSs to which it does not have direct access. The MMT considers $resources$ of FSs belonging to $List_{reach}$, and if they have enough resources to serve modules that are currently assigned to the current controller FS, it estimates the sojourn time of nth IoT device for those candidate FSs. Then, it returns the ID of the FS with sufficient resources and the maximum estimated sojourn time. It is important to note that assigning the controller role to a new FS with maximum sojourn time can reduce the number of possible future migrations, which leads to fewer service interruptions due to migration downtime. On the condition that no FSs of $List_{reach}$ contains enough resources, it returns the ID of FS with the maximum sojourn time. However, if $List_{reach}$ is empty, this method returns the ID of one of the FSs from $List_{unreach}$ randomly. Then, current controller FS sends a *NewControllerReq* message to $dest_{ID}$, containing the DAG of nth IoT device application $\mathcal{G}_n$, *mobilityInfo(n)*, and the current configuration of assigned servers X_n (lines 7-9).

When an FS receives *NewControllerReq* message, it adds the IoT device n to its *controllerList* to serve this IoT device as its new controller FS (lines 11-12). This new controller FS is responsible for the rest of migration management. It retrieves the current configuration X_n and the previous controller ID,

Algorithm 3: Migration Management Technique

```
Input : $RCM$: Received Control Message, $\mathcal{G}_n$: The DAG of $n$th IoT
        device, $mobInfo(n)$: The mobility data of the IoT device $n$,
        $X_n$: The configuration of assigned modules, $controller_{ID}$: ID
        of the controller, $List^n_{SFog}$: Sensed fog devices' List of IoT
        device $n$
1  switch $RCM$ do
2      case $MigrationInitiate$ do
3          $List^n_{SFog}$=$List^n_{SFog}$.remove($s_{ID}$)
4          $dest_{Id}$=MobilityAnalyzer(n,$mobInfo$,$List_{cl}$,$List^n_{SFog}$)
5          message.add($\mathcal{G}_n$,$mobInfo(n)$,$X_n$,$TO_n$,$SchS_n$)
6          message.type(NewControllerReq)
7          send($dest_{ID}$,message)
8          $controller_{pre}(n)$=true
9      end
10     case $NewControllerReq$ do
11         n=RCM.getIoTDevice
12         getcontrollerList().add(n)
13         $X_n$=RCM.getConfig(n)
14         $ID_{PreCon}$=RCM.getSourceAddr()
15         $List^{sorted}_{Cnts}$ =SortCntsSize($\mathcal{G}_n$, $Cnts^{ram}$)
16         $MapServer_{pre}$=FindPreServersConfig($X_n$)
17         for $t = 1$ to $|SchS_n|$ do
18             sendMigReqToServers($MapServer_{pre}$,$List^{sorted}_{Cnts}$,$SchS_{n.t}$)
19             WaitForServersNotifications()
20         end
21     end
22     case $MigrationReq$ do
23         $ReqInfo$=RCM.getInfo()
24         $Modules$= $ReqInfo$.getModules()
25         $S_R$=ReadyServers(this.getCMs(),this.getID(),this.getChildren())
26         if $!S_R$.isEmpty() then
27             for $i = 1$ to $Modules$.size() do
28                 $SortedCostList$=$\varnothing$
29                 for $j = 1$ to $S_R$.size() do
30                     $MigCostTemp$=CalMigCost($Modules_i$,$S_{R,j}$)
31                     CostList.update($S_{R,j}$,$MigCostTemp$)
32                 end
33                 $SortedCostList$=Sort(CostList)
34                 $Server_{ID}$=FindMigrationDestination($SortedCostList$)
35                 sendMigrationDestination($Modules_i$,$X_n$,$Server_{ID}$)
36             end
37         end
38         else
39             SendMigReqToServers(this.Parent(),$ReqInfo$)
40         end
41     end
42     case $MigrationDestination$ do
43         $v$=RCM.getModule()
44         $res_v$=calService($v$)
45         if $res_v \leq$ this.resources then
46             sendMigrationStart($v$,$FS^v_{pre}$,$FS^v_{new}$)
47             UpdateConfig($X_n$,$v$,$s_{ID}$)
48             NotifyController($v$,$s_{ID}$,$controller_{ID}$)
49         else
50             SendMMTFailureRecovery(n,v,$controller_{ID}$,$s_{ID}$)
51         end
52     end
53     case $StartMigration$ do
54         Migrate($v$,RCM.$FS^v_{new}$)
55         UpdateResoure($v$)
56         if $controller_{pre}(n)$ & MigrationFinish(n) then
57             $controller_{pre}(n)$=false
58             getControllerList().remove(n)
59         end
60     end
61     case MMTFailureRecovey do
62         MMTFailureRecovery($n$,$v$,$controller_{ID}$,$s_{ID}$)
63     end
64 end
```

ID_{PreCon}, from the received message RCM (lines 13-14). The *SortCntsSize* method descendingly sorts $Cnts$ based on their allocated runtime Ram $Cnts^{ram}$ (line 15). The background reason is the amount of dump and state to be transferred in the downtime is directly related to $Cnts^{ram}$ [31]. The migration of $Cnts$ with larger $Cnts^{ram}$ incurs higher cost in terms of migration time and energy (Eq. 27). Hence, to reduce the total migration cost, MMT gives higher priority to modules with heavier $Cnts^{ram}$ so that the migration decision can be made sooner, and they can be migrated before other modules. Next, *FindPreServersConfig* method retrieves assigned servers' specifications for all application modules and stores them in $MapServer_{pre}$ (line 16). The migration cost (Eq. 27) is defined as the maximum migration cost for each application module while considering X_n and its new configuration X'_n. The goal is to minimize this migration cost while it is subject to the condition that the new configuration X'_n provides better application execution cost or roughly the same with previous configuration X_n (Eq.29). So, the MMT retrieves modules of each schedule based on $SchS_n$ and send their corresponding information alongside $MapServer_{pre}$ and $List^{sorted}_{Cnts}$ to *sendMigReqToServers* method. It creates a list of modules based on the hierarchical layer on which modules are previously assigned. Modules of each hierarchical layer are also sorted based on allocated Ram size, obtained from $List^{sorted}_{Cnts}$. This method sends $MigrationReq$ messages alongside respective modules' information to FSs that are responsible for making the migration decision. As MMT acts in a distributed manner and FSs at each layer only has information about their parent, children, and CMs, migration decisions for modules of each layer are made by the new controller, its parent, or ancestors in the hierarchy. To illustrate, considering Fig. 1, we assume an IoT application has three modules in one of its schedules and two of them were previously assigned on FS (1,3) (prior controller), and one on FS (2,1). If we assume that the new controller is FS (1,4), it makes migration decision for modules that previously assigned on FS (1,3) while $par(1,4)$ (i.e., FS (2,3)) makes migration decision for the module that previously assigned on FS (2,1). After sending migration requests $migrationReq$, FS (1,4) waits to receive notifications and new configuration of modules for that schedule and then iterates over next schedules (lines 17-20).

When an FS receives *MigrationReq* message, the FS retrieves the information and forwarded modules from the received message (lines 23-24). Then, the list of ready servers S_R is created based on CMs, and children. If the S_R does not contain any available servers, all the modules are forwarded to the parent FS for making migration decision (line 39), while if it contains servers, it tries to minimize the migration cost based on the specification of available servers (line 26-37). This FS considers a list of $modules$, sorted descendingly based on $Cnts^{ram}$, for making migration decision. Hence, the migration of modules that incur higher migration costs in each schedule is performed with higher priority, leading to less overall migration costs in that schedule. Then, for each selected module, the migration cost is estimated and stored in the $CostList$ (line 29-32). The *Sort* method sorts the migration costs ascendingly so that servers with lower migration cost receives higher priority (line 33). Then, the $FindMigrationDestination$ method selects a new server for the module, considering *SortedCostList*, which minimizes the migration cost while it does not negatively affect the application's running cost. Hence, this method iterates over

$SortedCostList$, sorted ascendingly based on the migration costs, and selects the server that satisfies the Eq. 29 (line 34). Finally, the *sendMigrationDestination* method sends a $MigrationDestination$ message to the selected FS to check its resources and start the migration of the respective module.

The FS receiving *MigrationDestination* checks whether it has enough resources to serve the module v or not (lines 42-44). If this FS can serve the module v, it sends a *StartMigration* message to the FS^{v}_{pre} so that it can start the migration. Then, it updates the X_n with its s_{ID} and notifies the controller (lines 45-468). If it cannot serve this module due to any reason, it runs the *SendMMTFailureRecovery* method to send a failure message to the controller FS (lines 49-51).

The *MMTFailureRecovey* is working as the same as *DAPTFailureRecovey*. The only difference is that the migration cost in the MMT is obtained from Eq. 27 (lines 59-61).

Whenever an FS receives a *StartMigration* message, it starts the migration and then frees the previously assigned resources (lines 53-55). Moreover, if the FS was previously the controller for the nth IoT device, and it finishes the migration of all assigned modules belonging to that IoT device, the FS removes the nth IoT device from its *controllerList* (lines 56-59).

D. Complexity Analysis

The Time Complexity (TC) of the clustering phase (Algorithm 1) depends on the size of $List_{cl}$ and $List_{ch}$, and candidate parents in the immediate upper level for each FS. In the worst-case scenario, if we assume all FSs reside in one cluster and/or they have only one parent. Hence, the TC of *remove* method belonging to the *FogLeaving* and *FogFailureRecovery* is $O(F)$, and the TC of the *StartFogFailureRecovery* is $O(F)$. Moreover, the TC of *ParentSelection* method of *CandidParent* is $O(F)$ in the worst-case scenario if we assume one FS has $F-1$ candidate parent. Hence, the TC of the clustering step in the worst-case scenario is $O(F)$. Moreover, in the best-case scenario, the number of FSs in $List_{cl}$ and/or the size of the $List_{ch}$ is one, and the TC of the best-case is $O(1)$.

To find the TC of DAPT (Algorithm 2), we suppose that the size of the largest IoT application is K. So, in the worst-case scenario, the size of $U_{\mathcal{G}_n}$ is K. The *FindOrder* method finds the topological order of the DAG using BFS algorithm with the TC of $O(K+|\mathcal{E}|)$, in which $|\mathcal{E}|$ represents the number of data flows. In the dense DAG, the $|\mathcal{E}|$ is of $O(K^2)$. Moreover, the TC of *Sort* Algorithm is $O(FK^2)$ in the worst-case scenario. In the worst-case scenario, all FSs reside in one cluster and have enough resources for any requests. Hence, the worst-case TCs of *ClusterCheck*, *ReadyServers*, *FindMinCost*, and *DAPTFailureRecovery* are of $O(F)$, $O(F)$, $O(FK)$, and $O(FK)$, respectively. Hence, the worst-case TC of DAPT Algorithm is $O(FK^2+FK)$. In the best-case scenario, the DAG of the application can be sparse so that the TC of *FindOrder* and *Sort* algorithms become $O(K)$ and $O(1)$, respectively. Moreover, in the best-case scenario, the number of available servers in one cluster is one, and hence, TCs of *ClusterCheck*, *ReadyServers*, *FindMinCost*, and *DAPTFailureRecovery* are of $O(1)$, $O(1)$, $O(K)$, and $O(K)$, respectively. So, TC of DAPT in the best-case scenario is $O(K)$.

The TC of the *MigrationInitiate* from Algorithm 3 depends on the TC of *MobilityAnalyzer*. In the worst-case scenario, all the FSs reside in one cluster and the IoT device can sense all of them. So, the size of the list of sensed FSs $List^{n}_{SFog}$ is equal to F. Hence, in the worst-case, the TC of creating $List_{reach}$ and $List_{unreach}$ is of $O(F^2)$ while in the best-case scenario, it is of $O(F)$ when there is only one FS in the cluster. Moreover, the worst-case TC of finding maximum sojourn time is $O(F)$. So, the TC of *MigrationInitiate* in the worst-case is $O(F^2)$ while in the best-case, it is of $O(F)$. The TC of the *NewControllerReq* in the worst-case is $O(KLogK+FK)$ while TC of *NewControllerReq* in the best-case scenario is $O(KLogK)$ when there is only one FS in each cluster. The TC of *MigrationReq* in the worst-case scenario depends on the TCs of $CalMigCost$ and $Sort$ which are $O(FK)$ and $O(FKLogF)$ while in the best-case scenario they are O(K). The TC of *MigrationDestination* depends on the TC of $MMTFailureRecovery^{mig}$ and is of $O(F)$ at the worst-case and $O(1)$ in the best-case scenario. Therefore, the TC of the MMT in the worst-case scenario is $O(F^2+FK+KLogK+FKLogF)$ while in the best-case scenario is $O(KLogK)$.

Considering TCs of all methods, the TC of our technique in the worst-case scenario is $O(F^2+FK^2+FKLogF)$ while in the best-case scenario, it is $O(F+KLogK)$.

V. Performance Evaluation

In this section, the system setup and parameters, and detailed performance analysis of our technique, in comparison to its counterparts, are provided.

A. System Setup and Parameters

We extended the iFogSim simulator [10] for the implementation and evaluation of distributed mobility management, clustering, and failure recovery techniques. We used DAGs of two real-time applications, namely the Electroencephalography tractor beam game (EEGTBG) [10], [19] and ECG Monitoring for Health-care applications (ECGMH) [9] to create our DAGs. Both applications consist of a sensor and display modules that are placed in the IoT device (e.g., smartphone, wearable devices, etc). Other modules can be placed either on distributed FSs or CSs based on the distributed application placement decisions and/or the migration technique. Data transmission intervals for ECG and EEG sensors are 10ms and 15ms, respectively [1], [10]. Besides, we assume the amount of RAM allocated to each container at the runtime for state size is randomly selected from 50-75 MBytes [31]. The total amount of data to be transferred in the downtime (i.e., $dsize^{mig}$) is just a few MBytes [31], which is randomly selected from 5-10% of each container's allocated RAM in the runtime.

We simulate a 2km × 1km area, in which the coverage range of FSs situated in the first and second layers is assumed to be 200m and 400m, respectively. The system consists of one layer of IoT devices, three layers of heterogeneous FSs, and a layer [1], [7], [9]. The IoT device layer consists of 80 IoT devices, while the number of FSs in level 1, level 2, and level 3 are 30, 5, and 1, respectively. The computing power (CPU) of IoT devices is considered as 500 MIPS [33], while the computing power of level 1 FSs is randomly selected from [3000-4000] MIPS [33], [19]. Besides, the total computing power of level 2 FSs, level 3 FSs, and CS are considered as 8000 MIPS, 10000 MIPS, and 80000 MIPS, respectively [7], [9]. Besides, the latencies between IoT devices to level 1 FSs, level 1 FSs to level 2 FSs, level 2 FSs to level 3 FSs, and level 3 FSs to cloud servers are 5ms, 25ms, 50ms, and 150ms, respectively [1], [9], [7]. The upstream and downstream network capacity of IoT devices are 100 Mbps and 200 Mbps, respectively. The upstream, downstream, and clusterlink network capacity for FSs and the CSs are also considered to be 10 Gbps [7], [9]. Moreover, clusters can be formed among the level 1 and level 2 FSs with their in-range FSs of the same hierarchical layer. The communication latency among the FSs residing in level 1 clusters and FSs residing in level 2 clusters are [3-5] ms and [20-25] ms, respectively [1], [9]. The processing power consumption, idle power consumption, and transmission power consumption of IoT devices are 0.9W, 0.3W, and 1.3W, respectively [35], [2]. User trajectories are generated by a variation of the random walk mobility model [27], [20], in which each user selects a direction, chooses a destination anywhere toward that direction, and moves towards it with a uniformly random speed. The user arriving at the destination can choose a new random direction.

Table III: Evaluation Parameters

Parameter	**Value**
Simulation Time	100,200,300,400 (S)
Area	2km × 1km
Users' Speed	[0.5-4] m/s
Latency (ms)	
ECG Sensor Data Transmission Interval	10
EEG Sensor Data Transmission Interval	15
ECG and EEG Sensor ↔ IoT Device	2
IoT Device ↔ Level 1 FS	5
Level 1 FS ↔ Level 2 FS	25
Level 2 FS ↔ Level 3 FS	50
Level 3 FS ↔ Cloud	150
L1 Clusters	[3-5]
L2 Clusters	[20-25]

B. Performance Study

We conducted seven experiments evaluating system size analysis, average execution cost of tasks, cumulative migration cost, the total number of migrations, Total number of Interrupted Tasks (TIT) due to the migration, Failure recovery analysis, and optimality analysis. In the experiments, to obtain the weighted cost of placement and migration, the w_1 and w_2 are set to 0.5. To analyze the efficiency of our technique, we extended two other counterparts in the dependent category of fog computing proposals as follows:

- *MAAS*: This is the extended version of the technique called Mobility-Aware Application Scheduling (MAAS) [19] working based on edgeward-placement technique. The main concern of this edge-centric technique is to place dependent modules of IoT applications on remote servers based on their pre-known mobility pattern (i.e., source, destination, and the potential paths between them are known in advance) of users. In MAAS, if an FS cannot place modules on itself, the modules should be forwarded to the parent server for placement. We extended this technique to support the migration as the users move among remote servers in the runtime while considering the destination and potential paths are not priori-known.
- *Urmila*: This is the extended version of Ubiquitous Resource Management for Interference and Latency-Aware services (Urmila) [6] which proposes a mobility-aware technique for placement of dependent modules of IoT applications while mobility pattern of users are priori-known. In this technique, the central controller is placed in the highest level FS, and makes placement decisions for IoT applications consisting of dependent modules. We extended this technique so that the central controller helps remote servers to migrate dependent modules of applications as the IoT users move.

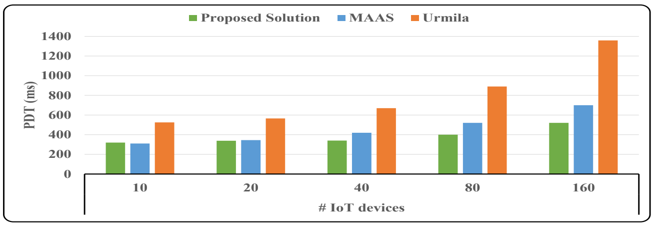
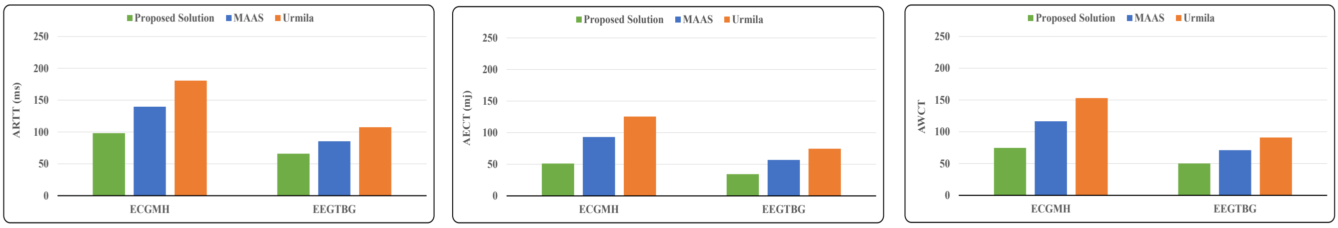


Figure 5: Placement Deployment Time (PDT)

1) System size analysis: In this experiment, we study the effect of number of IoT devices on the Placement Deployment Time (PDT). The PDT shows the period between the start of sending placement requests from IoT devices up to the time the deployment of application modules of IoT devices on FSs are finished. Obviously, the PDT includes the decision time in which FSs make placement decisions and the container startup cost on the servers. Regardless of the quality of solutions that each technique provides, the PDT helps to understand how long the IoT devices should wait until the service can start. In this experiment, the number of IoT devices is increased from 10 to 160 by multiplication of two. Although the number of IoT devices increases in this experiment, we fixed the number of FSs so that we can analyze how different techniques work when the number of placement requests increases significantly. Besides, it is clear that our technique, due to its distributed manner, can easily manage the increased number of placement requests when the number of FSs increases.

In Fig. 5, the PDTs of our proposed solution and MAAS are significantly lower than Urmila, specifically in a larger number of IoT devices. This latter is mainly because our solution and MAAS use a distributed placement engine while Urmila uses a centralized approach. When the placement decision engine receives incoming placement requests, it should make placement decisions and then manage the deployments of



(a) Average Response Time of Tasks (ARTT)

(b) Average Energy Consumption of Tasks (AECT)

(c) Average Weighted Cost of Tasks (AWCT)

Figure 6: Average execution cost of tasks

application modules in different servers according to solutions' configuration. In Urmila, all of the placement requests should be forwarded to the centralized entity, meaning that the number of arriving placement requests in the decision engine is larger than the distributed placement techniques. Hence, the processing of these requests on the centralized controller takes more time compared to the distributed placement engines, especially when the number of IoT devices increases. Moreover, our solution outperforms the MAAS since it tries to place more application modules in the lowest hierarchical layer, compared to MAAS, which incurs less deployment time.

2) Average execution cost of tasks: This experiment shows the average execution cost of tasks emitted from a sensor module until they arrive at actuator in 400 seconds of simulation.

As it can be seen from Fig 6, our proposed solution outperforms the MAAS and Urmila in terms of Average Response Time of Tasks (ARTT), Average Energy Consumption of Tasks (AECT), and Average Weighted Cost of Tasks (AWCT). In the MAAS, each FS, from the lowest to the highest hierarchical level, attempts to place modules on itself or forwards them to its parent server for the placement or handling of the migration process. Therefore, it does not consider other potential servers at the same hierarchical level, which incurs higher transmission and inter-nodal costs. The pure Urmila, on the other hand, does not migrate the application modules to servers that are closer to the moving IoT devices, and hence, the average execution cost of tasks, emitted from IoT devices, increases significantly. In our distributed technique, however, each FS considers potential servers at the same hierarchical level (for placement and migration) if those servers are among its CMs. In this way, we decrease the large search space of centralized techniques, while we use the benefits that servers at the same hierarchical level can provide. Also, since modules with higher costs have higher placement priority, the possibility of their placement on more suitable servers are higher compared to other modules. This latter leads to better placement decisions that minimize the cost of executing tasks. It is important to note that the average execution cost of the EEGTBG is lower than the ECGMH. It is because tasks' instruction number in the EEGTBG is lower than of ECGMH ones.

3) Total number of migrations: This experiment studies the total number of migrations that occurred during 400 seconds due to the IoT users' movement.

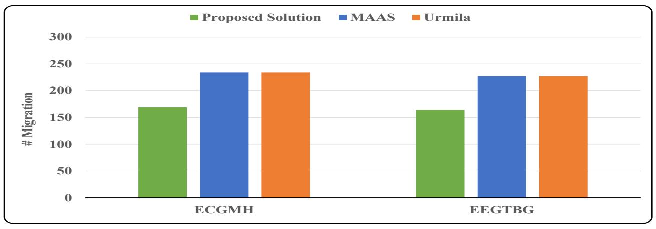
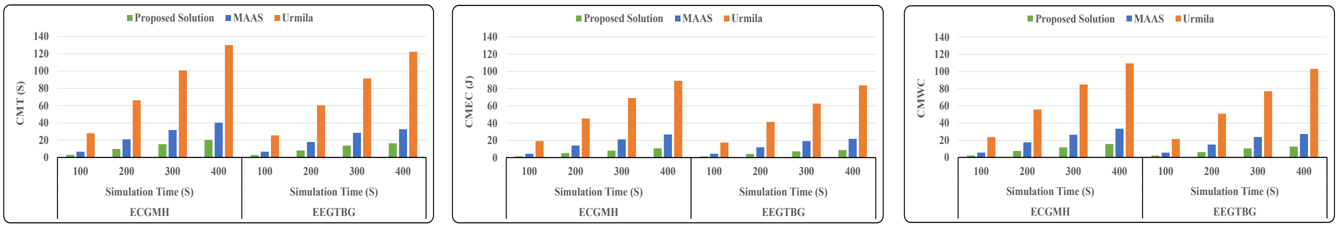


Figure 7: Total number of migrations

It can be seen from Fig. 7 that our technique leads to a smaller number of migrations in comparison to its counterparts. This is because our solution considers the current mobility information of IoT devices such as current speed and direction. Since the controller FS has coordinates of its CMs and current mobility information of leaving IoT devices (e.g., their average speed and their direction while in the range of the current controller FS), the serving FS can estimate a sojourn time for all candidate remote servers for the migration. Hence, by the migration of modules to the remote server with the highest sojourn time (in case sufficient resources are available), the number of possible migrations decreases. The extended MAAS and Urmila only try to reduce the migration cost by migrating modules to new remote servers, while they do not consider current mobility information of IoT devices and their sojourn time in remote servers. Hence, they may select remote servers in which the IoT devices stay only for a short period.

4) Cumulative migration cost: This experiment analyzes the Cumulative Migration Cost (CMC) of IoT devices for ECGMH and EEGTBG in different simulation times. The term cumulative refers to the aggregate migration cost of all IoT devices.

As Fig 8 shows, our solution outperforms its counterparts in terms of Cumulative Migration Time (CMT), Cumulative Migration Energy Consumption (CMEC), and Cumulative Migration Weighted Cost (CMWC) for both ECGMH and EEGTBG applications. As the simulation time increases, the cost of all techniques grows, however, Urmila experiences a faster increase in comparison to our solution and MAAS. This latter is because the Urmila's controller is placed at the highest hierarchical layer, which incurs significant inter-nodal and transmission cost when the controller manages migrations between the old and new remote servers in the downtime.



(a) Cumulative Migration Time (CMT)

(b) Cumulative Migration Energy Consumption (CMEC)

(c) Cumulative Migration Weighted Cost (CMWC)

Figure 8: Cumulative Migration Cost

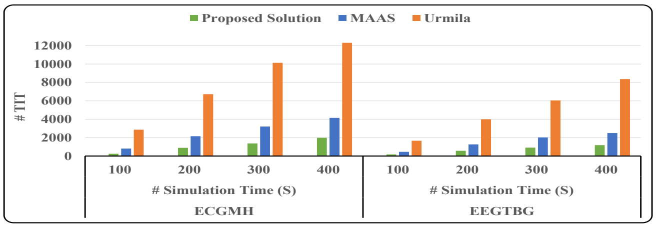


Figure 9: Total number of interrupted tasks

Besides, the migration cost of MAAS is more than our solution, since whenever the resources of controller finishes, the MAAS migrates the application modules to higher layers, and hence, the emitted tasks to/from those modules experience higher cost. Also, the total number of migrations in Urmila and MAAS are higher than ours, which apparently increases their cumulative migration costs. The slight difference between cost of ECGMH and EEGTBG is because the tasks generated from the ECGMH's modules are heavier than EEGTBG's ones in terms of their MI. So, the processing time of remaining instructions of tasks (i.e., $e_{n,i,j}^{ins,r}$) that migrated from old server to new server is higher for the ECGMH compared to the EEGTBG (in case the computing powers of old and new servers are roughly the same).

5) Total number of interrupted tasks (TIT): This experiment analyzes the Total number of Interrupted Tasks (TIT) in the downtime. During migration downtime, there is no active service provider for incoming tasks from the modules deployed on the IoT device for a while. Hence, service interruptions happen in the downtime, in which the generated tasks experience higher delays or even they can be discarded, compared to the tasks that are generated when there is no migration. The IoT users receive smoother results with lower TIT.

Fig. 9 presents the TIT of techniques for ECGMH and EEGTBG in different simulation times. It can be seen that our solution outperforms its counterparts in different simulation times for the ECGMH and EEGTBG. The migration time has a direct impact on the TIT, and the techniques with higher migration time lead to larger TIT. This latter is because as the migration time increases, the number of delayed (or even dropped) tasks grows faster. It can be seen from Fig. 9 that the Urmila results in larger TIT than two other techniques because of its higher migration time. Moreover, due to our smaller migration time, the TIT of our solution is smaller than other techniques for both ECGMH and EEGTBG applications. It is worth mentioning that the TIT of techniques for EEGTBG applications is smaller than of ECGMH ones. This latter is due to a higher data transmission interval for the EEG sensor in EEGTBG compared to the ECG sensor of ECGMH, which means that the number of emitted tasks per second for the EEGTBG application is smaller than the ECGMH application. Hence, applications with shorter task emission interval (here, the ECGMH application) suffer more from higher migration time.

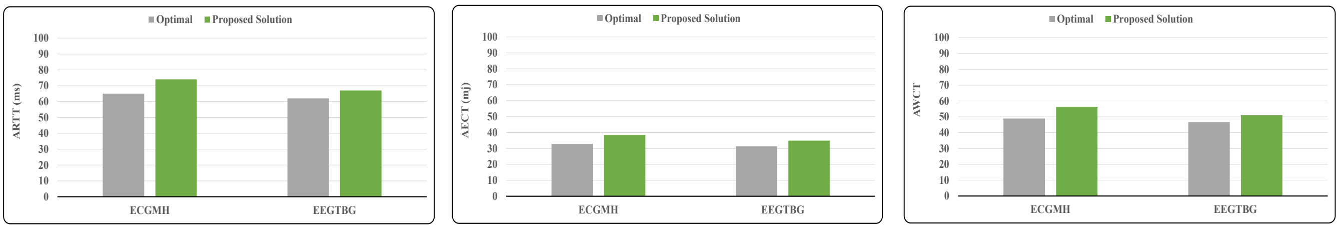
Table IV: Failure Recovery Analysis

Applications	Experiment	Techniques		
		Proposed Solution (FR Mode)	MAAS (No FR)	Urmila (No FR)
ECGMH	Total Number of Migrations	177	234	234
	Total Number of Interrupted Tasks	2095	4152	12302
EEGTBG	Total Number of Migrations	169	227	227
	Total Number of Interrupted Tasks	1228	2504	8361

6) Failure recovery analysis: In this experiment, we study the effect of the failure recovery method in the migration process. The MAAS and Urmila do not have any failure recovery methods and their results are just presented here for comparison purposes. The results of our technique with a failure recovery method (FR Mode) are presented in Table IV when there is a 5% probability of failure in the migration process.

Table IV illustrates that our technique with the failure recovery method (FR Mode) can recover from failures while it still outperforms its counterparts in terms of the total number of migrations and TIT. The obtained results of the average execution cost of tasks and cumulative migration cost in the FR Mode are roughly the same with the Non-FR Mode and they are not provided here. Since the Urmila and MAAS do not have any failure recovery methods, in case of any failures, their placement and/or migration process remains incomplete. However, in our technique, we embedded the failure recovery method for which it accepts a small overhead while it does not stop working if any failures occur.

7) Optimality analysis: In this experiment, we compare the performance of our proposed solution with the optimal values. To obtain the optimal results, we used an optimized version of the Branch-and-Bound algorithm to search all possible candidate configurations for application placement, in which the bounding function helps to faster prune the search space [36]. Since finding the optimal solution is very time consum-



(a) Average Response Time of Tasks (ARTT)

(b) Average Energy Consumption of Tasks (AECT)

(c) Average Weighted Cost of Tasks (AWCT)

Figure 10: Optimality analysis results

ing, in this experiment, we only consider 20 IoT devices in a hierarchical fog computing environment consisting of 15 candidate servers.

Fig.10 shows the results of optimality analysis in terms of Average Response Time of Tasks (ARTT), Average Energy Consumption of Tasks (AECT), and Average Weighted Cost of Tasks (AWCT). The results show that our solution has an average of 12% difference with the optimal results. However, considering the large number of FSs distributed in the proximity of IoT users, obtaining the optimal solutions, due to their large search spaces, is not practically possible, especially for real-time IoT applications.

VI. Conclusions and Future Work

We proposed a new weighted cost model for minimizing the overall response time and energy consumption of IoT devices in a hierarchical fog computing environment, in which heterogeneous FSs and CSs provide services for IoT devices. In order to enable collaboration among remote servers and provide better services for IoT applications, we proposed a dynamic and distributed clustering technique among FSs of the same hierarchical level. Considering the heterogeneous resources of remote servers and the dynamic nature of such computing environments, we also proposed a distributed application placement technique to place interdependent modules of IoT applications on appropriate remote servers while satisfying their resource requirements. Also, to manage potential migrations of IoT applications' modules among remote servers, due to IoT users' mobility, a distributed migration management technique is proposed. The main goal of this latter is to reduce the migration cost of IoT applications. Finally, we embedded light-weight failure recovery methods to handle possible unpredicted failures that may happen in such dynamic computing environments. The effectiveness of our technique is analyzed through extensive experiments and comparisons by the state-of-the-art techniques in the literature. The obtained results demonstrate that our technique improves its counterparts in terms of placement deployment time, average execution cost of tasks, the total number of migrations, cumulative migration cost of all IoT devices, and the total number of interrupted tasks due to migration.

As part of future work, we will extend our cost model to consider the energy consumption of servers and monetary cost. Moreover, we plan to consider different migration models such as pre-copy, post-copy, and hybrid, and analyze how they affect IoT applications with different resource requirements. Finally, we plan to integrate these techniques in real container-based distributed frameworks such as FogBus2 [14] framework to better analyze proposed techniques in real-world scenarios.

References

[1] R. Mahmud, K. Ramamohanarao, and R. Buyya, "Latency-aware application module management for fog computing environments," *ACM Transactions on Internet Technology (TOIT)*, vol. 19, no. 1, p. 9, 2018.

[2] M. Goudarzi, H. Wu, M. Palaniswami, and R. Buyya, "An application placement technique for concurrent iot applications in edge and fog computing environments," *IEEE Transactions on Mobile Computing*, vol. 20, no. 4, pp. 1298–1311, 2020.

[3] M. Goudarzi, M. Palaniswami, and R. Buyya, "A fog-driven dynamic resource allocation technique in ultra dense femtocell networks," *Journal of Network and Computer Applications*, vol. 145, p. 102407, 2019.

[4] F. Guo, H. Zhang, H. Ji, X. Li, and V. C. Leung, "An efficient computation offloading management scheme in the densely deployed small cell networks with mobile edge computing," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2651–2664, 2018.

[5] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2015.

[6] S. Shekhar, A. Chhokra, H. Sun, A. Gokhale, A. Dubey, and X. Koutsoukos, "Urmila: A performance and mobility-aware fog/edge resource management middleware," in *2019 IEEE 22nd International Symposium on Real-Time Distributed Computing (ISORC)*. IEEE, 2019, pp. 118–125.

[7] M. Taneja and A. Davy, "Resource aware placement of iot application modules in fog-cloud computing paradigm," in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE, 2017, pp. 1222–1228.

[8] A. Kiani, N. Ansari, and A. Khreishah, "Hierarchical capacity provisioning for fog computing," *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 962–971, 2019.

[9] S. Pallewatta, V. Kostakos, and R. Buyya, "Microservices-based iot application placement within heterogeneous and resource constrained fog computing environments," in *Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing*, 2019, pp. 71–81.

[10] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, "ifogsim: A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments," *Software: Practice and Experience*, vol. 47, no. 9, pp. 1275–1296, 2017.

[11] L. Yang, H. Zhang, X. Li, H. Ji, and V. C. Leung, "A distributed computation offloading strategy in small-cell networks integrated with mobile edge computing," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2762–2773, 2018.

[12] S. Jošilo and G. Dán, “Computation offloading scheduling for periodic tasks in mobile edge computing,” *IEEE/ACM Transactions on Networking*, vol. 28, no. 2, pp. 667–680, 2020.

[13] S. Wang, Y. Guo, N. Zhang, P. Yang, A. Zhou, and X. S. Shen, “Delay-aware microservice coordination in mobile edge computing: A reinforcement learning approach,” *IEEE Transactions on Mobile Computing*, 2019, (in press).

[14] Q. Deng, M. Goudarzi, and R. Buyya, “Fogbus2: a lightweight and distributed container-based framework for integration of iot-enabled systems with edge and cloud computing,” in *Proceedings of the International Workshop on Big Data in Emergent Distributed Environments*, 2021, pp. 1–8.

[15] A. Machen, S. Wang, K. K. Leung, B. J. Ko, and T. Salonidis, “Live service migration in mobile edge clouds,” *IEEE Wireless Communications*, vol. 25, no. 1, pp. 140–147, 2017.

[16] S. Wang, R. Urgaonkar, T. He, K. Chan, M. Zafer, and K. K. Leung, “Dynamic service placement for mobile micro-clouds with predicted future costs,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 4, pp. 1002–1016, 2016.

[17] T. Ouyang, Z. Zhou, and X. Chen, “Follow me at the edge: Mobility-aware dynamic service placement for mobile edge computing,” *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 10, pp. 2333–2345, 2018.

[18] M. Adhikari, S. N. Srirama, and T. Amgoth, “Application offloading strategy for hierarchical fog environment through swarm optimization,” *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4317–4328, 2019.

[19] L. F. Bittencourt, J. Diaz-Montes, R. Buyya, O. F. Rana, and M. Parashar, “Mobility-aware application scheduling in fog computing,” *IEEE Cloud Computing*, vol. 4, no. 2, pp. 26–35, 2017.

[20] S. Wang, R. Urgaonkar, M. Zafer, T. He, K. Chan, and K. K. Leung, “Dynamic service migration in mobile edge computing based on markov decision process,” *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 1272–1288, 2019.

[21] Z. Wang, Z. Zhao, G. Min, X. Huang, Q. Ni, and R. Wang, “User mobility aware task assignment for mobile edge computing,” *Future Generation Computer Systems*, vol. 85, pp. 1–8, 2018.

[22] C. Yang, Y. Liu, X. Chen, W. Zhong, and S. Xie, “Efficient mobility-aware task offloading for vehicular edge computing networks,” *IEEE Access*, vol. 7, pp. 26 652–26 664, 2019.

[23] Z. Liu, X. Wang, D. Wang, Y. Lan, and J. Hou, “Mobility-aware task offloading and migration schemes in scns with mobile edge computing,” in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2019, pp. 1–6.

[24] C. Zhu, G. Pastor, Y. Xiao, Y. Li, and A. Ylae-Jaeaeski, “Fog following me: Latency and quality balanced task allocation in vehicular fog computing,” in *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 2018, pp. 1–9.

[25] C. Zhang and Z. Zheng, “Task migration for mobile edge computing using deep reinforcement learning,” *Future Generation Computer Systems*, vol. 96, pp. 111–118, 2019.

[26] F. Yu, H. Chen, and J. Xu, “Dmpo: Dynamic mobility-aware partial offloading in mobile edge computing,” *Future Generation Computer Systems*, vol. 89, pp. 722–735, 2018.

[27] Y. Sun, S. Zhou, and J. Xu, “Emm: Energy-aware mobility management for mobile edge computing in ultra dense networks,” *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2637–2646, 2017.

[28] Q. Qi, J. Wang, Z. Ma, H. Sun, Y. Cao, L. Zhang, and J. Liao, “Knowledge-driven service offloading decision for vehicular edge computing: A deep reinforcement learning approach,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4192–4203, 2019.

[29] D. Wang, Z. Liu, X. Wang, and Y. Lan, “Mobility-aware task offloading and migration schemes in fog computing networks,” *IEEE Access*, vol. 7, pp. 43 356–43 368, 2019.

[30] H. Sami, A. Mourad, and W. El-Hajj, “Vehicular-obus-as-on-demand-fogs: Resource and context aware deployment of containerized micro-services,” *IEEE/ACM Transactions on Networking*, vol. 28, no. 2, pp. 778–790, 2020.

[31] C. Puliafito, C. Vallati, E. Mingozzi, G. Merlino, F. Longo, and A. Puliafito, “Container migration in the fog: a performance evaluation,” *Sensors*, vol. 19, no. 7, p. 1488, 2019.

[32] M. Goudarzi, Z. Movahedi, and M. Nazari, “Mobile cloud computing: a multisite computation offloading,” in *2016 8th International Symposium on Telecommunications (IST)*. IEEE, 2016, pp. 660–665.

[33] X. Xu, Q. Liu, Y. Luo, K. Peng, X. Zhang, S. Meng, and L. Qi, “A computation offloading method over big data for iot-enabled cloud-edge computing,” *Future Generation Computer Systems*, vol. 95, pp. 522–533, 2019.

[34] W. Zhang, J. Chen, Y. Zhang, and D. Raychaudhuri, “Towards efficient edge cloud augmentation for virtual reality mmogs,” in *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, 2017, pp. 1–14.

[35] K. Kumar and Y.-H. Lu, “Cloud computing for mobile users: Can offloading computation save energy?” *Computer*, no. 4, pp. 51–56, 2010.

[36] M. Goudarzi, M. Zamani, and A. T. Haghighat, “A fast hybrid multi-site computation offloading for mobile cloud computing,” *Journal of Network and Computer Applications*, vol. 80, pp. 219–231, 2017.

15th International Symposium Advances in Artificial Intelligence and Applications

THIS track is a continuation of international AAIA symposiums, which have been held since 2006. It aims at establishing the synergy between technical sessions, which encompass wide range of aspects of AI. With its longest-tradition threads, such as WCO focusing on Computational Optimization, it is also open to new initiatives categorized with respect to both, the emerging AI-related methodologies and practical usage areas. Nowadays, AI is usually perceived as closely related to the data, therefore, this track's scope includes the elements of Machine Learning, Data Quality, Big Data, etc. However, the realm of AI is far richer and our ultimate goal is to show relationships between all of its subareas, emphasizing a cross-disciplinary nature of the research branches such as XAI, HCI, and many others.

AAIA'21 brings together scientists and practitioners to discuss their latest results and ideas in all areas of Artificial Intelligence. We hope that successful applications presented at AAIA'21 will be of interest to researchers who want to know about both theoretical advances and latest applied developments in AI.

Topics

Papers related to theories, methodologies, and applications in science and technology in the field of AI are especially solicited. Topics covering industrial applications and academic research are included, but not limited to:

- Decision Support
- Machine Learning
- Fuzzy Sets and Soft Computing
- Rough Sets and Approximate Reasoning
- Data Mining and Knowledge Discovery
- Data Modeling and Feature Engineering
- Data Integration and Information Fusion
- Hybrid and Hierarchical Intelligent Systems
- Neural Networks and Deep Learning
- Reinforcement Learning
- Bayesian Networks and Bayesian Reasoning
- Case-based Reasoning and Similarity
- Web Mining and Social Networks
- Business Intelligence and Online Analytics
- Robotics and Cyber-Physical Systems
- AI-centered Systems and Large-Scale Applications
- AI for Combinatorial Games, Video Games and Serious Games
- Evolutionary Algorithms and Evolutionary Computation
- Computational Optimization (14th Workshop WCO'21)

Track Chairs

- **Ślęzak, Dominik,** University of Warsaw, Poland
- **Matwin, Stan,** Dalhousie University, Canada

Program Chairs

- **Świechowski, Maciej,** QED Software, Poland
- **Sosnowski, Łukasz,** Dituel, Poland

Program Committee

- **Agre, Gennady,** Bulgarian Academy of Sciences, Bulgaria
- **Bianchini, Monica,** University of Siena, Italy
- **Calpe Maravilla, Javier,** University of Valencia, Spain
- **Chelly, Zaineb,** Université de Versailles Saint-Quentin en Yvelines UFR des Sciences, France
- **Cyganek, Bogusław,** AGH University of Science and Technology, Poland
- **Dey, Lipika,** TCS Innovation Lab Delhi, India
- **Düntsch, Ivo,** Brock University, Canada
- **Girardi, Rosario,** UNIRIO, Brazil
- **Grabowski, Adam,** University of Bialystok, Poland
- **Ignatov, Dmitry,** National Research University Higher School of Economics, Russia
- **Jaromczyk, Jerzy,** University of Kentucky, United States
- **Jin, Xiaolong,** Chinese Academy of Sciences, China
- **Kasprzak, Włodzimierz,** Warsaw University of Technology, Poland
- **Kayakutlu, Gulgun,** Istanbul Technical University, Turkey
- **Lingras, Pawan,** Saint Mary's University, Canada
- **Loukanova, Roussanka,** Stockholm University, Sweden; and Institute of Mathematics and Informatics Bulgarian Academy of Sciences, Bulgaria
- **Markowska-Kaczmar, Urszula,** Wroclaw University of Technology, Poland
- **Matson, Eric,** Purdue University, USA
- **Matwin, Stan,** Dalhousie University, Canada
- **Menasalvas, Ernestina,** Universidad Politécnica de Madrid, Spain
- **Meneses, Claudio,** Universidad Católica del Norte, Chile
- **Moshkov, Mikhail,** King Abdullah University of Science and Technology, Saudi Arabia
- **Mozgovoy, Maxim,** University of Aizu, Japan
- **Myszkowski, Paweł,** Wroclaw University of Science and Technology, Poland

- **Pataricza, András,** Budapest University of Technology and Economics, Hungary
- **Peters, Georg,** Munich University of Applied Sciences & Australian Catholic University, Germany & Australia
- **Po, Laura,** Universitá di Modena e Reggio Emilia, Italy
- **Porta, Marco,** University of Pavia, Italy
- **Przybyła-Kasperek, Małgorzata,** University of Silesia, Poland
- **Raghavan, Vijay,** University of Louisiana at Lafayette, USA
- **Ramanna, Sheela,** University of Winnipeg, Canada
- **Rauch, Jan,** University of Economics, Prague, Czech Republic
- **Reformat, Marek,** University of Alberta, Canada
- **Schaefer, Gerald,** Loughborough University, England
- **Sikora, Marek,** Silesian University of Technology, Poland
- **Stanczyk, Urszula,** Silesian University of Technology, Poland
- **Stoean, Catalin,** University of Craiova, Romania
- **Subbotin, Sergey,** Zaporozhye National Technical University
- **Szczech, Izabela,** Poznan University of Technology, Poland
- **Unland, Rainer,** University of Duisburg-Essen, ICB, Germany
- **Weber, Richard,** University of Chile, Chile
- **Verstraete, Jörg,** Systems Research Insitute, Polish Academy of Sciences, Poland
- **Zakrzewska, Danuta,** Institute of Information Technology Technical University of Lodz, Poland
- **Zdravevski, Eftim,** Ss.Cyril and Methodius University, Macedonia
- **Zaineb Chelly,** Aberystwyth University, Wales
- **Zielosko, Beata,** University of Silesia, Poland

DOI: 10.15439/2021F29
ISSN 2300-5963 ACSIS, Vol. 25

An efficient approach towards the generation and analysis of interoperable clinical data in a knowledge graph

Jens Dörpinghaus*, Vera Weil‡, Sebastian Schaaf†, Tobias Hübenthal‡
* German Center for Neurodegenerative Diseases (DZNE), Bonn, Germany, Email: jens.doerpinghaus@dzne.de
† Fraunhofer Institute for Algorithms and Scientific Computing, Schloss Birlinghoven, Sankt Augustin, Germany
‡ Department of Mathematics and Computer Science, University of Cologne, Germany

***Abstract*—Knowledge graphs have been shown to play an important role in recent knowledge mining settings, for example in the fields of life sciences or bioinformatics. Contextual information is widely used for NLP and knowledge discovery tasks, since it highly influences the exact meaning of expressions and also queries on data.**

The contributions of this paper are (1) an efficient approach towards interoperable data, (2) a runtime analysis of 14 real-world use cases represented by graph queries and (3) a unique view on clinical data and its application, combining methods of algorithmic optimisation, graph theory and data science.

I. Introduction

Personalized, or more precisely, stratified medicine aims for matching certain risk groups and possibly yet unknown subgroups to treatments, ultimately optimizing patients' responses, mainly to available drugs. This explicitly includes strategies beyond guidelines and off-label usage of substances in order to increase the effect and/or to decrease undesired side effects. For this purpose, collected primary data of the examined persons have to be linked with data from secondary sources like publications or databases in an application-oriented way. [1]

Considering both amount and complexity of input data, at least early steps of data processing require computer methods and thus machine-readable data. While these early steps are highly automated and deeply influence subsequent analyses, proper modeling of data and derived knowledge is key for each input, analysis and output layers of a system supporting decisions by human expert users. Considering data of interest to be highly heterogeneous as well as evolving and growing over time, the abstraction to semantic entities and relations appears favorable. Thus, graphs appear as a feasible basis for modeling both data and metadata. Moreover, external resources could be linked to such a knowledge graph. Finally, for querying contents modeled in such way, highly generic and powerful graph algorithms are available.

But how can clinical patient data from a database be efficiently stored as a knowledge graph using a suitable data schema? And how can queries be generated afterwards using the data linked in this way? What is the runtime of an application-related query for suitable literature for a given patient?

In this work, building on the already stored *PubMed* data from [2], a model will be presented that efficiently embeds the collected primary data into structured, domain-specific environments, in this case ontologies. The thereby generated knowledge graph will then be used to examine individual queries in terms of efficiency. See figure 1 for an illustration schema. For the purpose of illustration of real-world data a small section of the underlying graph in Neo4j can be seen in Fig. 2.

The contributions of this paper are an efficient approach towards interoperable data, a runtime analysis of 14 real world use cases represented by graph queries and a unique view on clinical data and its application combining methods of algorithmic optimisation, graph theory and data science. This paper is divided into seven sections. After an introduction, the second section gives a brief overview over the state of the art and related work. The third section describes the theoretical and practical background and the methods used for our novel approach. Therefore, we will refer to both knowledge graphs and dedicated algorithms. In the fourth section, we present our optimization approaches to tackle the interoperability challenges within our use cases. The fifth section covers applications from real-world use cases like query finding, with the experimental results on both artificial and real-world scenarios in the subsequent section. After that, we present a detailed evaluation in section six. Our conclusions and outlooks are drawn in the final section. We will propose a

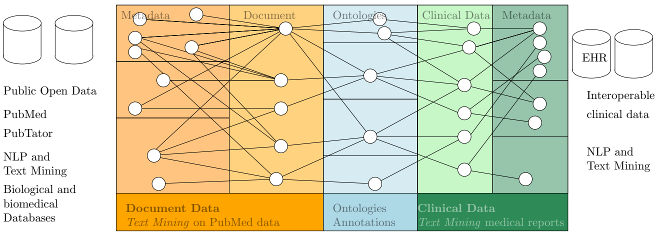


Fig. 1. Illustration of some knowledge graph layers found in the testing environment. Here, we can see document data extracted using Text Mining on PubMed Data with their metadata (e.g. authors, publication venues, publication date) and named entities from different ontologies. They form the interoperability layer to the clinical data. In the background we use medical reports analyzed with the help of text mining methods.

novel algorithmic approach which presents promising performance. The results show a significant improvement for new algorithms for knowledge discovery on clinical data.

II. Related work

In scientific research, expert systems provide users with several methods for knowledge discovery. They are widely used to find relevant or novel information. A popular example in biomedical research is to try to find molecular pathways; controlled reaction mechanisms within biological organisms, which might be misregulated in pathogenic states. Obviously, understanding these cascades, their players and relations to diseases is key to design and apply drugs in a targeted way.

Being confronted with patients' clinical data and with expert knowledge in the back of their minds, clinical researchers usually consider an initial idea and start integrating external content such as scientific papers. The most common approach is inquiring with a search engine about some terms to find closely related information. Effectively, users most frequently query for additional documents or patient files to adjust the search query. Similarly, for a given set of documents or patients the question might be on commonalities considering a certain topic. Both approaches are heavily related to the context of data points, see for example [3] for PubMed data. Topic labelling – or cluster labelling – is constantly being explored in several research fields.

In principle, the way external data sources and manually curated data are integrated is key. Although several commercial solutions exist, Fakhry et al. state that the "adoption and extension of such methods in the academic community has been hampered by the lack of freely available, efficient algorithms and an accompanying demonstration of their applicability using current public networks." [4]. This and the emerging improvements on large-scale Knowledge Graphs and machine learning approaches are the motivation for our novel approach on semantic Knowledge Graph embeddings for biomedical research utilising data integration with linked open data. Several similar approaches (often in the context of drug-repurposing) such as Bio2RDF [5], hetionet [6], or OpenPHACTS [7] have already been described. Our approach is more focussed on integrating the literature itself in a FAIR [10] and open knowledge graph, which is also accessible as a public resource.

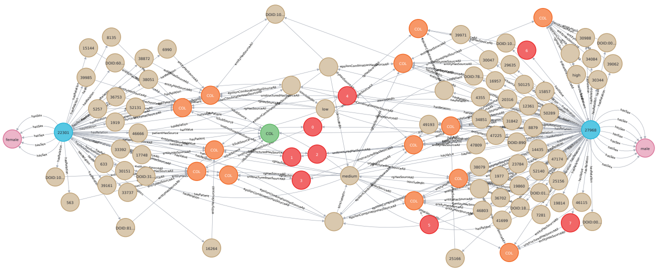


Fig. 2. Example illustration of the proposed knowledge graph centered around "Alzheimer's disease". We find different patients (blue nodes), values (red nodes) and ApoE allele combinations (orange nodes). The light brown nodes refer to entities, for example from Disease Ontology. These are highly important for the interoperability of the knowledge graph.

In recent decades the field of natural language processing (NLP) and knowledge discovery as well as data mining and the management of information systems as the related fields are emerging. It is not exactly the focus of this paper, but since we are relying on data obtained from biomedical texts, we should note that several authors like Manning et al. [8] or Clarc et al. [9] give an overview about the algorithmic part of computational linguistics and NLP. In addition there is a constant interest in using graphs for these problems, see [10].

III. Background

Using graph structures to house data carries several advantages for the integration of knowledge and its targeted re-extraction. According to their generic character such integrative knowledge graphs are important for the life sciences, medical research and associated fields, not least supporting their interconnection on a formal level. Considering systems medicine applications, knowledge graphs provide grounds for holistic approaches unraveling disease mechanisms. In these and other common settings pathway databases play an important role. As a basis, biomedical literature and text mining are used to build knowledge graphs, see [11]. As part of the studies on integrative data semantics within clinical research, data on patients suffering from certain diseases have been collected by various institutions. In our case, the data was available in the form of a NoSQL Mongo database. In addition, several databases and ontologies can implicitly form a knowledge graph. For example Gene Ontology, see [12], DrugBank, see [13] or [14] cover large amounts of relations and references which other fields can refer to.

In [2] we collected 27 real world questions and queries in scientific projects to test the performance and output of the knowledge graph. We could show that the performance of several queries was very poor and some of them even did not terminate. In order to identify limitations and understand the underlying problems, we carried on with our work. The testing system is based on *Neo4j* and holds a dense large-scale labeled property graph with more than 71M nodes and 850M edges. They are based on biomedical knowledge graphs as described in [11].

The more specific data model for clinical research used in this work initially contains general variables about a patient, e.g., site of measurement (*site*), sex (*sex*), and genotypes (ApoE). Additionally, a variety of neuropsychological testing results (NPT), recorded disorders (disturbances), laboratory measurements (LAB), liquid biopsy markers (LIQ), spinocerebellar ataxia characterisations (SCA) and diagnoses (both ICD10-encoded and free text) are given as variables.

Due to the sensitivity of personal clinical data, a sample of artificial data is used within this work. They do not cover the full data schema presented later (see Fig. 6). However, the simplified model is sufficient to present the principle of the work, and can be easily transferred or extended to a full data set using the data schema.

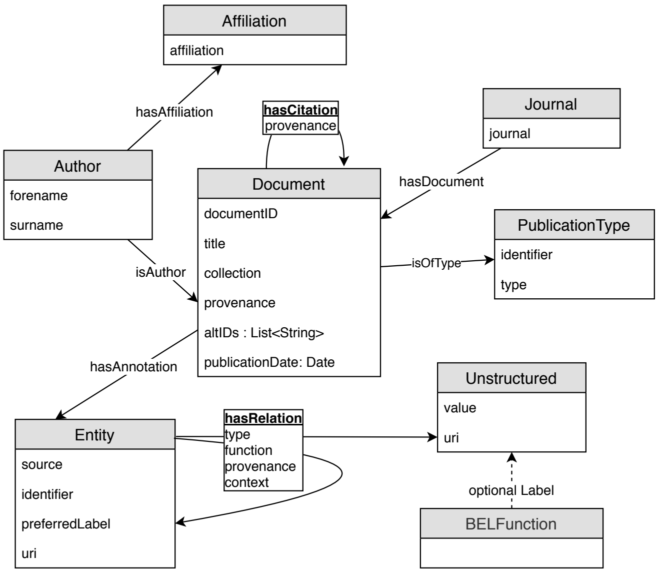


Fig. 3. Formal data schema of the knowledge graph of the *PubMed* database.

This work is also based on the Paper *Towards context in large scale biomedical knowledge graphs* [2], in which an efficient polyglot persistence design for storing and querying a knowledge graph based on the documents contained in the *PubMed* database is presented on the basis of a created data schema. There, based on questions from the biomedical domain, a knowledge graph is created by resorting to the data schema given in Fig. 3. [2]

Data schema and practical implementation are based on clinical questions, which are the result of several interviews with employees from the biomedical field. They form the basis for the knowledge graph requirements.

IV. Optimization approaches

A. Data Schema and Knowledge Graph Foundations

Based on the given data, we need to tackle the challenge of how to arrange data points in a suitable schema and thus store them efficiently in a knowledge graph so that the clinical questions can be answered using graph queries. Simultaneously, the later connection to the graph of the *PubMed* database in [2] needs to be solved. For this purpose, we use the schema represented in Fig. 3. Extended with data from clinical studies an enrichment with context data is also possible, applying a mapping between the entities and several ontologies. Here, a mapping means a function $M : E(D) \to E(O)$, where $E(D)$ describes the entities of the given data and $E(O)$ the entities of the used ontologies. Thus, the mapping consists of edges that create a logical relation between biomedical data and lexicons in the knowledge graph, see [2].

First, the used classes of the model presented in Fig. 6 should be introduced. The initial and central class is the patient itself. It does not contain any other attributes except for an ID, as those are outsourced for the reasons mentioned above.

The next class to be considered is the patients' gender which is represented by the node `sex`. Another class is the ApoE risk type. It describes a genetic condition that is closely associated with risks for Alzheimer's disease. For each patient, either a 2-tuple $(\epsilon i, \epsilon j)$, $i, j \in \{1, 2, 3, 4\}$ or two 2-tuples $(rs429358, rs7412)$ of SNPs, which in turn form a $\epsilon i, i \in \{1, 2, 3, 4\}$, are given. The individual alleles in this case again consist of a tuple of SNPs, which can be either of type C or T. For illustration, the small model is shown in Fig. 4.

This construction results in the class of the risk group, which can have different patient-specific expressions. These expressions are implicitly given by the risk types. Three different categories are distinguished with the following designations:

- low-risk: patients whose risk type corresponds to a tuple $(\epsilon i, \epsilon j)$, $i, j \in \{1, 2, 3\}$.
- medium-risk: patients whose risk type corresponds to a tuple $(\epsilon i, \epsilon j)$ with $i = 4, j \in \{1, 2, 3\}$ or $i \in \{1, 2, 3\}, j = 4$.
- high-risk: patients whose risk type corresponds to the tuple $(\epsilon 4, \epsilon 4)$.

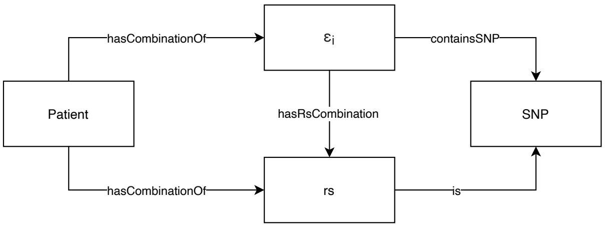


Fig. 4. Structure of ϵi-tuples and rs-codes. For further explanaition regarding the biology behind this concept, see [15]

In our case, the ϵ tuple is used for the artificial data. However, the model can of course be applied to the alleles as well, since there is a unique bijective mapping between the set of alleles relevant to the ApoE and the ϵ-tuples. [15]

Next, we consider measurements of clinical trials and studies. These include, as already mentioned in. section III, LAB, LIQ, NPT, ApoE, SCA, and diagnoses. They are collected in the class of `attributes` and belong to a parent category, *topic* stored under `unstructured`. Values are assigned to the `attributes` as part of the patient examinations. Like the *topics*, these are created as separate objects of the `Unstructured` class and linked to the `attributes` class via relations. The values themselves are not initially associated with any entity. Another class `unit` is created, which stores the associated unit for each measured value. This, as mentioned above, again allows for a wider range of viewing and interrogation options for the graph. In addition, one can connect the units to a suitable ontology, for example UCUM (Unified Code for Units of Measure, [16]), using appropriate relations.

Furthermore, one or more diagnoses can be assigned to a patient. These are also defined as a separate class and are stored as a *diagnosis code*. This matches the *diagnosis codes*

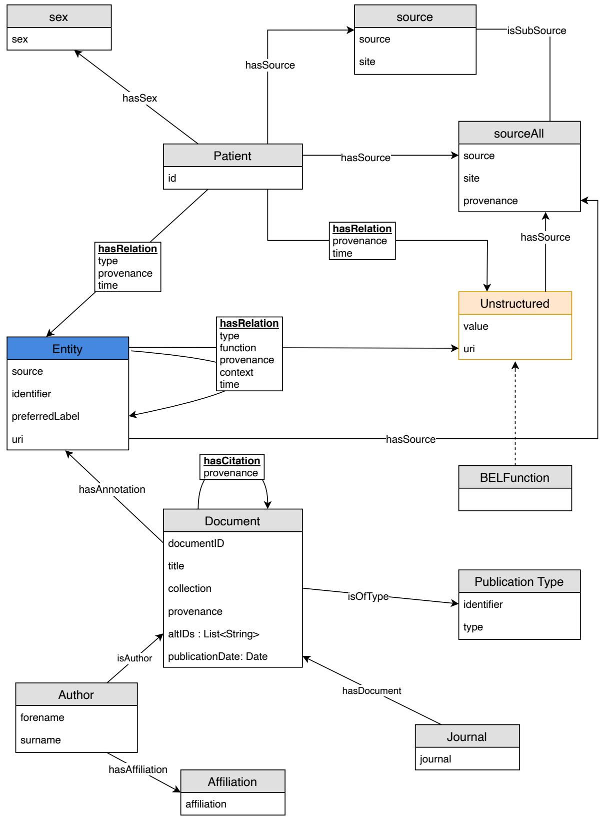


Fig. 5. Formal data schema of the knowledge graph combining clinical data and document data.

of the *Disease Ontology* in its descriptive usage, so it is also used by default in subject-related literature. [17]

In addition, we use the class `time`. Every measurement and every examination is provided with a timestamp. Thus, a temporal hierarchy can be determined for any subset of the total data and, under certain circumstances, even trends regarding the course of the disease can be identified. Commonly, the time stamps describe the date of an examination, but might also retroactively refer to the date of the first occurrence of a symptom or an event.

The last classes used are `source` and `sourceAll`. Here, `source` serves as the source of a dataset of a specific clinical institute of the DZNE. The location where the data was collected is also stored. This enables to locally delineate the data from each other and allows for combining datasets of different origins in one graph without losing their affiliation.

The class `sourceAll`, inheriting from `source`, is almost identical to the latter, but has the additional attribute *provenance*. This makes it possible to unambiguously define and record the relationships between different instances of classes within a given data set. The classes mentioned above have in different relations to one another. However, the outsourcing of attributes and the resulting increase in the number of nodes also increases the number of edges. In a directed graph $G = (V, E)$ with a number of nodes $n = |V|$, the number of edges in the worst case is $|E| = n \cdot (n-1)$, since each node $v \in V$ can have an edge to any other node $u \in V, u \neq v$.

The naming of the edges follows the Dublin Core standards, see [18]. These are simple standards of the Dublin Core Metadata Initiative for data formats of documents or objects. The vocabulary listed there has partially been replaced by more domain-specific expressions, while preserving the basic structure. As an example the term *hasFormat* is replaced by the more adequate *hasSex* which keeps the original structure and represents the `patient-sex` relationship.

In addition to the name or label of an edge, further information is required. Each relation of two classes receives a timestamp, which is stored in the attribute *time*. This can be given explicitly by a concrete date or also implicitly by the relationship of the classes and relations to each other. In addition, the edges receive an attribute *provenance*. This correlates with the attribute of the same name of the `sourceAll` node and assigns an internal membership to relations between several classes. Thus, the *provenance* attribute, whether within a node class or as an attribute of a relation, prevents data ambiguity.

These considerations yield the schema in Fig. 6. However, we need to create a more generic schema for the graph database. Therefore, we extend the schema from [11] by combining all blue marked entities from Fig. 6. Thus, the schema in Fig. 5 is obtained.

B. Creating interoperable data

We use the bulk import function of *Neo4j* to load the data. For this, we converted the input data into CSV files. This import consists of the following steps:

Algorithm 1 INTEROPERABLE-DATA

```
Require: import file $f$
Ensure: CSV export $c$
    createPathEnvironment($c$)
    createStandardNodes()
    impCSVFile($f$)
    writeNodes($f$,$c$)
    writeEdges($f$,$c$)
    writeHeaders($c$)
    return $c$
```

The method `createPathEnvironment(`c`)` creates the path environment for the export c, `createStandardNodes()` creates the standard nodes (e.g. for `sex`). They can be outsourced of the main methods below as they are static and do not depend on the input data.These first steps can be done in linear time.

The main method is `impCSVFile(`f, c`)`. First, it gets the import file f containing the clinical data. Then, for each entry all necessary nodes and relations are created while avoiding duplicates with the help of lists and sets. Regarding the time complexity of the look-up function, the latter is far superior to the former: Sets in Python are implemented as hash tables using keys as indices and therefore provide an average look-up time of $\mathcal{O}(1)$ instead of lists which need $\mathcal{O}(n)$ (see [19].

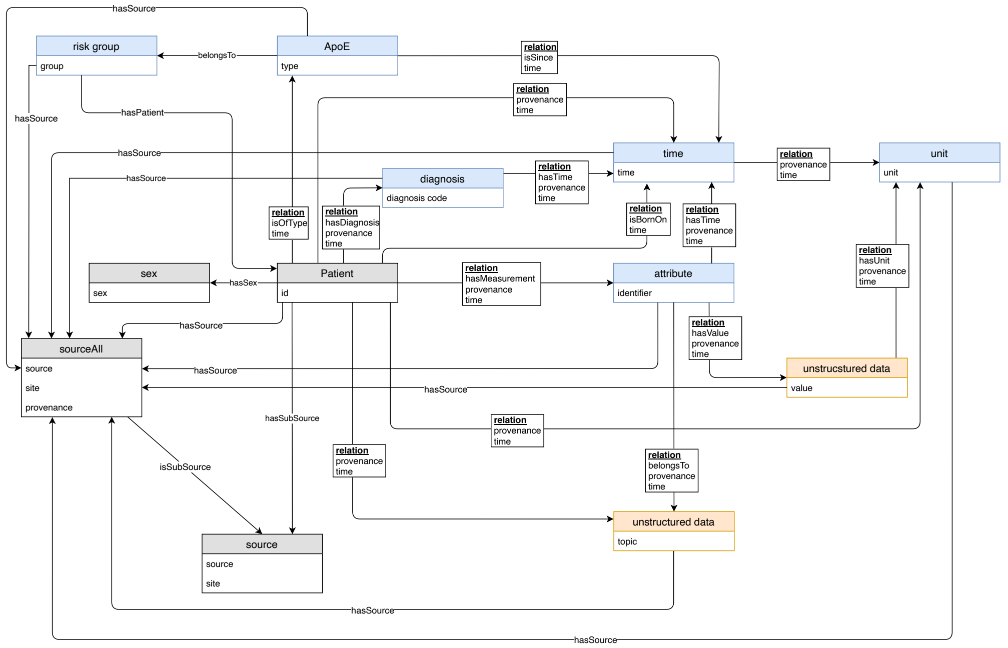


Fig. 6. Detailed data schema deduced from the IDSN data model. In orange, the node class `Unstructured` represents unstructured data considered complementary contextual information provided for entities and patients.

`writeNodes`(f,c) and `writeEdges`(f,c) then print the CSV files from the nodes and edges created beforehand. Both simply go through a list of stored nodes which takes $\mathcal{O}(n)$ time. Lastly, `writeHeaders` creates the necessary header files for the bulk import which happens in $\mathcal{O}(1)$ (see [20]).

With out first approach using lists to avoid duplicate nodes and edges we achieved a total time complexity of $\mathcal{O}(n^2)$ where n represents the size of our import data f. When replacing lists with sets we could lower the total time complexity to $\mathcal{O}(n)$. The explicit runtimes shown in Fig. 11 in section VI-D prove the theoretical difference.

V. Use-cases and Graph Queries

We obtain our use-cases from clinical questions. These are ordered and categorized within this section in order to later analyze their efficiency in VI and to consider comparative values by means of complexity theory. Preliminary work has been carried out in [21] and [22]. There, biomedical questions are examined and optimized as well. For this purpose, six different literature sources are used, some of which cite different and some common categories or classes for graph-based queries. In [2], a hierarchy for the classification of the queries is created with reference to fitting literature sources and using the author's own criteria.

This section focuses on the individual biomedical issues. They were collected through interviews with clinical and biomedical professionals. The process is similar to the one in [2] First the input and output for all questions is described. Since the graph contains only a subset of the actual intended data, the original clinical questions have to be replaced. However, the categories of the queries, assigned as proposed in [2], shall be maintained. From a biomedical perspective, these questions may not make sense or might not be of interest; however, from a computer science perspective they are isomorphic to the original ones. The questions are numbered in the listing so that they correlate with the later queries.

In spite of no biological question to base upon, queries 13-15 have been added to test the algorithms provided within *Neo4j*. The associated queries formulated in *Cypher* can be found in table I.

VI. Evaluation

The queries created in V are now to be applied to the graph and evaluated with respect to their runtimes. For this purpose, the actual runtime in *Neo4j* is measured for multiple executions and related to the time complexity of the algorithms. The queries are presented according to their categories. Here, the numbering is done following table I.

A. RPQ

The simplest query in the class of Graph Navigation Queries is whether a certain path exists in the graph, see [22]. Path queries specified with regular expressions are commonly referred to as *Regular Path Queries (RPQ)*, see [23].

Most of the queries are RPQs (Q1, Q2, Q3, Q5, Q7, Q11, Q12). According to [24], these have a polynomial runtime due to transformation into a non-deterministic finite automaton.

Query 1 refers to the question *For which patients do complete neuropsychological tests exist?*. This question can be substituted by *Which patients have the most distinct HGNC values?*. The corresponding Cypher query is:

TABLE I
CLINICAL QUESTIONS, THEIR COMPLEXITY CLASS AND REPLACEMENTS FOR TESTING PURPOSE. HERE DC REFERS TO *Degree Centrality*, SP TO *Shortest Path* AND BC TO *Betweenness Centrality*.

	Class	Question	Replacement
1	RPQ	For which patients do complete neuropsychological tests exist?	Which patients have the most distinct HGNC values?
2	RPQ	Which measurement values are most common in the context of {diagnosis1}?	Which patients are found most often in the context of a risk group {RiskGroup1}?
3	RPQ	Which measurements are collected at the same time?	Which HGNC values are most commonly collected during a certain visit {visit1}?
4	CRPQ	What does the chronological order of the measured values of entity {entity1} and risk group {RiskGroup2} for a patient {patient1} look like?	What does the chronological order of the measured HGNC values {entity1} and risk group {RiskGroup2} for a patient {patient1} look like?
5	RPQ	How many patients received a diagnosis within two days of their visit?	How many patients received a diagnosis on their first visit?
6	CRPQ	Which patients diagnosed with {diagnosos2} underwent neuropsychological testing {number1} days beforehand?	Which patients diagnosed with {diagnosis2} at visit {visit2} had the HGNC value {HGNC_value} at their previous visit?
7	RPQ	Do people of age {age1} come for examination more often than others?	Do people of sex {sex1} come for examination more often than others?
8	CRPQ	Which entity {entity2} do patients without any diagnosis have in common?	Which patients are diagnosed with exactly {number} distinct diagnoses and to which risk groups do they belong?
9	DC	How often does an allel tuple {allel tuple} appear amongst all patients?	Which risk group is most common amongst all patients?
10	ECRPQ	What literature {literature1} can be found for patient {patient2} diagnosed with {diagnosis3}?	Which HGNC values {HGNC_value2} can be found for patient {patient2} diagnosed with {diagnosis3}?
11	RPQ	How many patients underwent neuropsychological testing {npt1} and at the same time have laboratory value {LAB_value1}?	How many patients are diagnosed with {diagnosis4} and at the same time have an HGNC value {HGNC_value}?
12	RPQ	How many Patients suffer from disturbance {disturbance1} and what sex are they?	How many patients are diagnosed with {diagnose5} and what sex are they?
13	SP	-	What is the shortest path between entity {entity4} and entity {entity5} and what is on this path?
14	BC	-	Which patient connects entities most strongly?

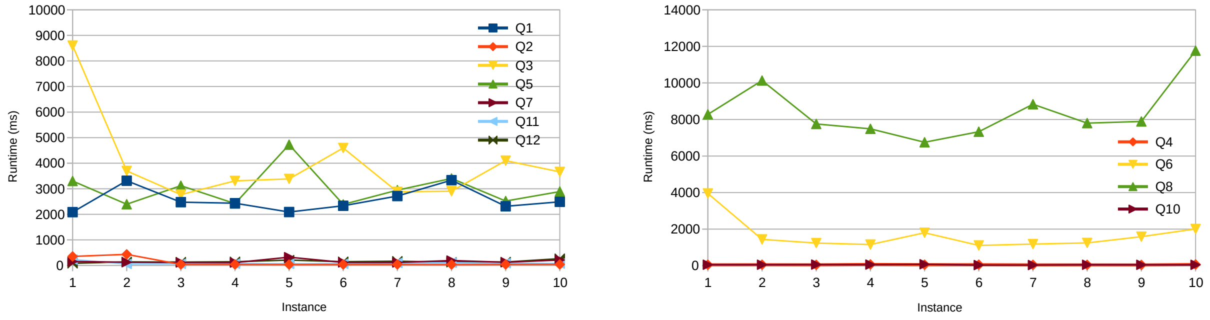


Fig. 7. Runtimes of different RPQ queries (left) and runtimes of (E)CRPQs queries (right) in milliseconds. Mean values are 2558.5 (Q1), 109.8 (Q2), 3990.7 (Q3), 3010.5 (Q5), 160.3 (Q7), 90.9 (Q11) and 154.5 ms (Q12) for RPQ and 53.6 (Q4), 1671.7 (Q6), 8402.6 (Q8) and 52.2 ms (Q10).

(Q1)
```
MATCH (e:Entity {source:'HGNC'})
<- [:hasRelation]-(p:Patients) RETURN
p.patient AS Person, COUNT(DISTINCT e)
AS number ORDER BY number DESC LIMIT 10
```

As another example query 3 answers the question *Which measurements are collected at the same time?*. It can be substituted by *Which HGNC values are most commonly collected during a certain visit {visit1}?*. The corresponding Cypher query is:

(Q3)
```
MATCH (e:Entity {source:'HGNC'}) <-
[:hasRelation]-(p:Patients)-[:hasValue]
-> (v:Unstructured {value:'2'}) RETURN
e.preferredLabel AS HGNC_Wert, COUNT(e) AS
number ORDER BY number DESC LIMIT 10
```

The respective average runtimes of each query can be seen in 7. As we can see, we have slower (Q1, Q3 and Q5) and faster queries(Q2, Q7, Q11 and Q12).

As discussed in [22], queries become slower with increasing number of queried attributes and used relations. Here, we first consider the queried attributes. However, these do not differ fundamentally for the two groups of fast and slow RPQs. They vary between one and three, but several attributes are needed even for the faster running queries. The comparison of the relations leads to more conspicuousness: Regarding the number of relations used by the slow (Q3 and Q4) and the fast

Queries (Q7 and Q12), on first sight, there is no difference. They all make use of two relations. But when looking at the nodes used within the queries a notable difference can be found:One of the nodes used in Q7 and Q12 ist `sex` which only has two possible specifications: male and female. The queries Q3 and Q4 have a similar structure, but instead of using `sex` they use the far larger class `entity` which holds more than 50,000 nodes. The former class contains only two distinct nodes (male and female), while the latter class contains over 50,000 nodes. Thus, many more nodes need to be checked, which explains the speed difference.

B. CRPQ and ECRPQ

Conjunctive queries and *RPQ*s can be combined in the class *CRPQ*, see [23], which then can be extended further by the extended CRPQs which include the possibility to specify path variables and even allows the output of a query to be a path. According to the schema presented in [22] they both are subproblems of pattern matching in graphs. We consider both CRPQs (Q4, Q6, Q8) and ECRPQ (Q10, see Fig. 8) together.

For example, Query 4 is answering the question *What does the chronological order of the measured values of entity entity1 and risk group RiskGroup2 for a patient patient1 look like?* This question can be substituted with *What does the chronological order of the measured HGNC values entity1 and risk group RiskGroup2 for a patient patient1 look like?* The Cypher query can be formulated as follows:

(Q4)
```
MATCH (u:Unstructured) <-
[:hasValue]-(p:Patients {patient:
'22504'}) - [:hasRelation] -> (e:Entity
{source: 'HGNC'}) MATCH (r:RiskGroups)
- [:hasPatient] -> (p) RETURN
e.preferredLabel AS HGNC, r.riskgroup AS
RiskGroup, u.value AS VisitNo ORDER BY
u.value DESC
```

Question 10 (ECRPQ) answers the question *What literature {literature1} can be found for patient {patient2} diagnosed with {diagnosis3}?* In our testing environment we can substitute this query with *Which HGNC values {HGNC_value2} can be found for patient {patient2} diagnosed with {diagnosis3}?*. The Cypher query is expressed as follows:

```
10: MATCH p=(e1:Entity
{preferredLabel:"Alzheimer's disease"})
<- [:hasRelation] -(A:Patients {patient:
"12864"}) - [:hasRelation] -> (e2:Entity
{source:"HGNC"}) RETURN p
```

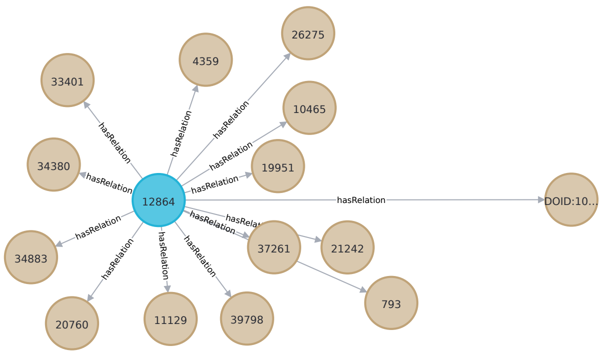


Fig. 8. Example output of ECRPQ query Q10.

The results and runtimes can be found in Fig. 7.

According to [25], the class of CRPQ is $\mathcal{NP}$-complete, so it can probably not be solved efficiently. The figure mentioned above clearly shows this fact because, besides very good results for Q4 and Q10, one also finds very poor runtimes for Q6 and Q8. The last-mentioned query immediately catches the eye, since, according to I, it covers several of the above-mentioned aspects. It is the only one of the (E)CRPQs presented here that works globally, accesses four different attributes and uses two different relations. Q6 does not search globally, indeed only locally, but it also has two different edge types, each of which occurs in both conjugate subquerys, and even queries four different attributes.

In particular Q4 is interesting. The query only searches locally, but queries five different attributes and uses three different relation types. Nevertheless it has quite a low runtime and the structure of the graph might offer an explanation for this seeming discrepancy. A directed graph $G = (V, E)$ with $n = |V|$ nodes can have up to $n \cdot (n-1)$ edges under the assumption that there are no duplicate edges, because each node v can have an edge (v, u) to any other node $u \in V, v \neq u$. Looking at the source files used here, the low density of the graph is immediately noticeable. An example of this is provided by the input CSV file, which contains about 30,000 patients, but at most six visits per patient, which in turn include fewer than 20 relations. So there are no more than 120 edges per patient. Looking at the whole graph, this ratio can also be seen when importing into the database: with half a million nodes, only slightly more than 2.6 million edges are created. Thus, it is noticeable in Q4 that the small portion of data that the query looks at is rather sparse. There is very little data on a patient, so only a very limited number of possibilities needs to be considered. This may explain the quick response of the database.

Lastly, we take a look at the ECRPQ Q10. It uses three different attributes but only one edge type. Here, we find the same result as previously discussed for Q4: only a very limited amount of data is available for the patient, which is likely to have a significant impact on the speed of the query.

C. Other results

Finally, we take a closer look at the algorithms used. All three are algorithms integrated into *Neo4j* and provided via the Graph Data Science 1.1.3 plug-in (Q9, Q13, see Fig. 9 and Q14). First, we explore Q9 with Degree Centrality.

(Q9)
```
CALL gds.alpha.degree.stream({
nodeProjection: ['Entity','Patients'],
relationshipProjection: 'hasPatient'
}) YIELD nodeId, score RETURN
gds.util.asNode(nodeId).identifier AS
name, score AS numberOfPatients ORDER BY
```

`numberOfPatients DESC LIMIT 3` Since the incident edges of each node are counted, for a dense graph $G = (V, E)$ we obtain a complexity of $\mathcal{O}(|V|^2)$ in the worst case, i.e. every node $v \in V$ is incident to every other $u \in V, u \neq v$. In [22], it is already shown that the algorithms implemented in *Neo4j* have such a high running time on large networks that they become practically almost useless. However, the runtimes and the average of query Q9 shown in Fig. 10 initially show otherwise for Degree Centrality. Once again, the fact that the graph is very sparse comes into play. The nodes have only a few direct neighbors and this has a strong effect on the runtime of the algorithm. Here, we use Degree centrality for a node with only a few different specifications. Thereby, the long runtime can be circumvented by constructing three individual queries for the different risk groups instead of using the algorithm in the first place.

Similarly, Q14 calculates the betweenness centrality:

(Q14)
```
CALL gds.graph.create
('myUndirectedGraph',
["Patients","Entity"], {hasRelation:
{orientation: 'UNDIRECTED'}})
CALL gds.alpha.betweenness.stream
('myUndirectedGraph') YIELD
nodeId, centrality RETURN
gds.util.asNode(nodeId).preferredLabel
AS entityLabel, centrality AS number ORDER
BY number DESC LIMIT 10
```

Q13 computes a shortest path. The Cypher query calls the function `shortestPath`:

(Q13)
```
MATCH(entity1:Entity {identifier:
'DOID:0040005'}) MATCH(entity2:Entity
{identifier: '41022'}) CALL
gds.alpha.shortestPath.stream({startNode:
entity1, endNode: entity2, nodeProjection:
'*', relationshipProjection:
{all:{type: '*', orientation:
'UNDIRECTED'}}}) Yield nodeId, cost RETURN
gds.util.asNode(nodeId), cost
```

The runtime is shown in Fig. 10. Q13 has a long runtime which is not surprising, as centrality measures for knowledge graphs are quite complex. While several efficient algorithms have been proposed, see [26], and some more specific problems are known to be $\mathcal{NP}$-hard. Here, good examples are Group Closeness Maximization (GCM), see [27] or the Maximum Betweenness Centrality, see [28].

There are several algorithms to compute shortest paths in graphs. The algorithm used in *Neo4j* is said to be a variant of Dijkstra's algorithm, which has a time complexity of $\mathcal{O}(|V| \cdot log|V| + |E|)$ [22]. For the given query, the algorithm runs very fast on the existing graph. However, as shown in the source above, the runtime grows tremendously with graphs less favourable to the algorithm, so the result obtained here should be viewed with caution. Finally, the problem becomes clear with the third algorithm. According to [29], Betweenness Centrality has a running time of $\mathcal{O}(|V|^3)$, but this can be improved with Brandes' algorithm to a time complexity of $\mathcal{O}(|V| \cdot |E|)$. According to the documentation of *Neo4j*, this is also the algorithm used there (cf. [30]). The query executed here, as shown by the average and the exact values in Fig. 10, ran for several hours before coming to a result. This not only stands in stark contrast to the other two algorithms, especially the one for shortest paths, but also renders them almost unusable for practical applications.

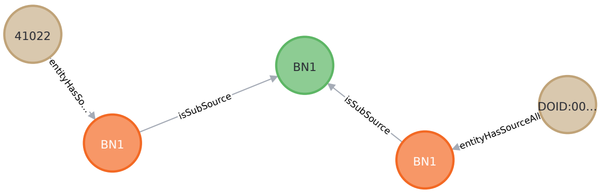


Fig. 9. Example output of shortest path query Q13.

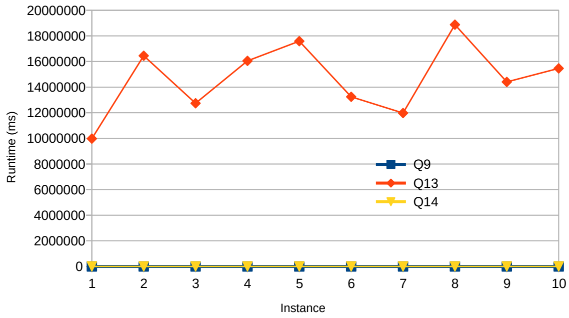


Fig. 10. Runtimes of different queries (Q9 degree centrality, Q13 betweenness centrality shortest path and Q14) in miliseconds. The mean values are 350.3 (Q9), 14,677,102.6 (Q13) and 1,118.5 ms (Q14).

D. Import

As introduced above, we proposed two different algorithmic approaches to import the data and generate interoperable data. For testing purposes, we created three different test sets: Two small data sets with 1,300 and 36,000 records as well as a large set with 135,000 data records.

See Fig. 11 for a detailed runtime overview. The optimized approach using sets is faster and thus more competitive for large data sets.

VII. Conclusion and Outlook

Here, we presented a novel approach that annotates clinical research data with contextual information. The result is a knowledge graph representation of data, the context graph. It contains computable statement representation. We discuss the impact of this novel approach using 14 real-world use cases and graph queries. This graph allows to compare research data records from different sources as well as the selection of relevant data sets using graph-theoretical algorithms. See Fig. 12 for an illustration of Alzheimer's data points.

This proof of concept of a biomedical knowledge graph combines several sources of data by relating their contextual

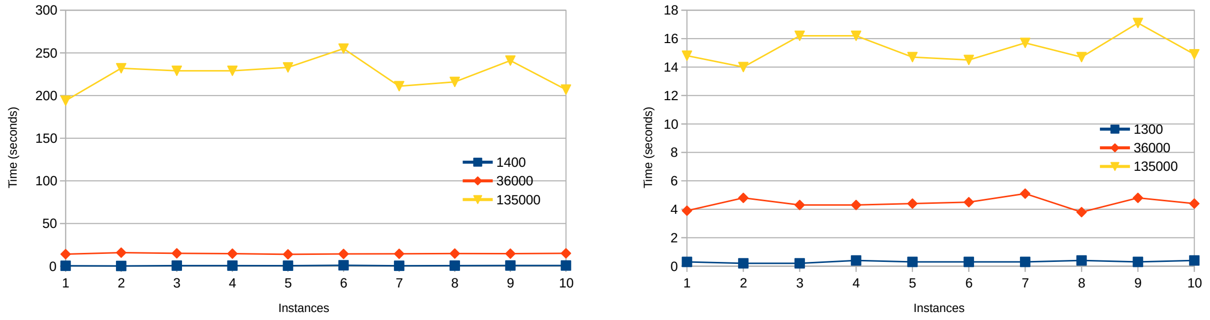


Fig. 11. Runtime of INTEROPERABLE-DATA, see Algorithm 1, with lists (left) and sets (right) with different import data size (1300–135000 data points). The speedup factor is between 2 (for small instances) and 13 (for large instances).

data to one another. We processed data from clinical research, biomedical publications and presented a generic and efficient approach towards interoperable data.

Furthermore, we discussed the runtime analysis of 14 real-world use cases represented by graph queries. As stated in previous works and discussed in our paper, performance for some semantic queries remains a major problem due to the massive latency for requesting detailed data points. Thus, the next step is to integrate the results presented in [22] in our information systems to improve the practical execution times for those and similar queries.

Storing and querying a giant knowledge graph as a labeled property graph is still a technological challenge. Here, we demonstrate how our data model is able to support the understanding and interpretation of biomedical data, especially in the context of clinical trials. We presented several real-world use cases that utilize our massive, generated knowledge graph. To date, we restricted our work to some smaller subgraphs. We plan to integrate these graphs into larger knowledge graphs, for example interaction networks. That will improve this unique view on clinical data and its application combining methods of algorithmic optimisation, graph theory and data science.

Considering the integration of further related biomedical knowledge resources, a variety of highly specific, field-dependent databases appears available at hand. For example, imaging techniques such as DICOM files from radiology reports are a promising extension for the future. Files of this format offer far more than just the image data, but contain a variety of additional (meta-)information concerning the patient, the applied imaging techniques, the circumstances of the examination, affiliated publications and much more. These meta-data could be reorganized in form of a graph and then added to the already existing model. However, it is important to find a suitable sequence for importing data, avoiding node ambiguity, e.g. considering already existing patient IDs.

While our proof of concept is both functional and generic, extending the knowledge graph to further fields of research, e.g. towards genomic and pharmacologic information or demographic background data, is feasible and just a matter of modelling connectors to the relevant sources. Moreover, these occasionally bulk data might be queryable on the fly through interfaces dynamically translating graph (sub-)queries into e.g. SQL. Despite considerable increases in expected running times, such partially distributed approaches might be favorable over integrated, warehouse-like solutions provisioning giant, largely unused graphs. However, efficient generation of remote queries and the re-integration of their results would obviously be key.

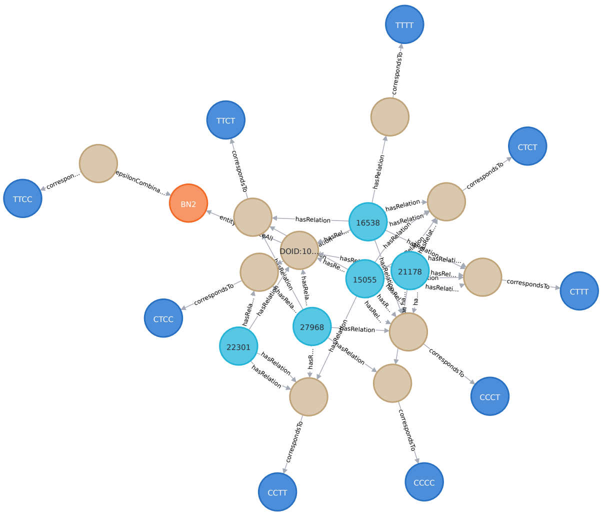


Fig. 12. Example nodes with output surrounding `preferredLabel:"Alzheimer’s disease"`. It describes the data foundation for novel approach that annotates clinical research data with contextual information. The result is a knowledge graph representation of data, the context graph. It contains computable statement representation.

REFERENCES

[1] “Integrative Daten-Semantik für die Neurodegenerationsforschung https://www.idsn.info/de/idsn.html,” Juli 2020. [Online]. Available: https://www.idsn.info/de/idsn.html

[2] J. Dörpinghaus, A. Stefan, B. Schultz, and M. Jacobs. (2020) Towards context in large scale biomedical knowledge graphs. [Online]. Available: http://arxiv.org/abs/2001.08392

[3] C. S. Burns, R. M. Shapiro, T. Nix, J. T. Huber *et al.*, “Examining medline search query reproducibility and resulting variation in search results,” *iConference 2019 Proceedings*, 2019.

[4] C. T. Fakhry, P. Choudhary, A. Gutteridge, B. Sidders, P. Chen, D. Ziemek, and K. Zarringhalam, “Interpreting transcriptional changes using causal graphs: new methods and their practical utility on public networks,” *BMC bioinformatics*, vol. 17, no. 1, pp. 1–15, 2016.

[5] F. Belleau, M.-A. Nolin, N. Tourigny, P. Rigault, and J. Morissette, “Bio2rdf: towards a mashup to build bioinformatics knowledge systems,” *Journal of biomedical informatics*, vol. 41, no. 5, pp. 706–716, 2008.

[6] D. S. Himmelstein, A. Lizee, C. Hessler, L. Brueggeman, S. L. Chen, D. Hadley, A. Green, P. Khankhanian, and S. E. Baranzini, “Systematic integration of biomedical knowledge prioritizes drugs for repurposing,” *Elife*, vol. 6, p. e26726, 2017.

[7] L. Harland, “Open phacts: A semantic knowledge infrastructure for public and commercial drug discovery research,” in *International Conference on Knowledge Engineering and Knowledge Management*. Springer, 2012, pp. 1–7.

[8] C. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[9] A. Clark, C. Fox, and S. Lappin, *The handbook of computational linguistics and natural language processing*. John Wiley & Sons, 2013.

[10] H. Mirisaee, E. Gaussier, C. Lagnier, and A. Guerraz, “Terminology-based text embedding for computing document similarities on technical content,” *arXiv preprint arXiv:1906.01874*, 2019.

[11] J. Dörpinghaus and A. Stefan, “Knowledge extraction and applications utilizing context data in knowledge graphs,” in *2019 Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 2019, pp. 265–272.

[12] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig *et al.*, “Gene ontology: tool for the unification of biology,” *Nature genetics*, vol. 25, no. 1, p. 25, 2000.

[13] D. S. Wishart, Y. D. Feunang, A. C. Guo, E. J. Lo, A. Marcu, J. R. Grant, T. Sajed, D. Johnson, C. Li, Z. Sayeeda *et al.*, “Drugbank 5.0: a major update to the drugbank database for 2018,” *Nucleic acids research*, vol. 46, no. D1, pp. D1074–D1082, 2017.

[14] K. Khan, E. Benfenati, and K. Roy, “Consensus qsar modeling of toxicity of pharmaceuticals to different aquatic organisms: Ranking and prioritization of the drugbank database compounds,” *Ecotoxicology and environmental safety*, vol. 168, pp. 287–297, 2019.

[15] “SNPedia https://www.snpedia.com/index.php/APOE,” Juli 2020.

[16] “UCUM- The Unified Code for Units of Measure http://unitsofmeasure.org,” Juli 2020.

[17] J. Hastings, *The Gene Ontology Handbook*. Springer, 2017, ch. Primer on Ontologies, pp. 3–13.

[18] “Dublin Core Metadata Initiative https://www.dublincore.org/specifications/dublin-core/,” Juli 2020.

[19] M. Gorelick and I. Ozsvald, *High Performance Python: Practical Performant Programming for Humans*. O’Reilly Media, 2014. [Online]. Available: https://books.google.de/books?id=bIZaBAAAQBAJ

[20] “Import in Neo4j https://neo4j.com/docs/operations-manual/current/tools/neo4j-admin-import/,” Juli 2020.

[21] J. Dörpinghaus and A. Stefan, “Knowledge extraction and applications utilizing context data in knowledge graphs,” in *2019 Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 2019, pp. 265–272.

[22] J. Dörpinghaus and A. Stefan, “Optimization of Retrieval Algorithms on Large Scale Knowledge Graphs,” 2020.

[23] P. T. Wood, “Query Languages for Graph Databases,” *SIGMOD Rec.*, vol. 41, no. 1, pp. 50–60, apr 2012. [Online]. Available: http://doi.acm.org/10.1145/2206869.2206879

[24] A. O. Mendelzon and P. T. Wood, “Finding Regular Simple Paths in Graph Databases,” *SIAM Journal on Computing*, vol. 24, no. 6, pp. 1235–1258, 1995. [Online]. Available: https://doi.org/10.1137/S009753979122370X

[25] P. T. Wood, “Query Languages for Graph Databases,” *SIGMOD Rec.*, vol. 41, no. 1, p. 50–60, Apr. 2012. [Online]. Available: https://doi.org/10.1145/2206869.2206879

[26] F. Grando, D. Noble, and L. C. Lamb, “An analysis of centrality measures for complex and social networks,” in *2016 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2016, pp. 1–6.

[27] C. Chen, W. Wang, and X. Wang, “Efficient maximum closeness centrality group identification,” in *Australasian Database Conference*. Springer, 2016, pp. 43–55.

[28] M. Fink and J. Spoerhase, “Maximum betweenness centrality: approximability and tractable cases,” in *International Workshop on Algorithms and Computation*. Springer, 2011, pp. 9–20.

[29] M. Fink, “Zentralitätsmaße in komplexen Netzwerken auf Basis kürzester Wege,” Master’s thesis, Julius-Maximilians-Universität Würzburg: Institut für Informatik, 2009.

[30] “Neo4j Betweenness Centrality https://neo4j.com/docs/graph-data-science/current/algorithms/betweenness-centrality/,” Juli 2020.

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 69–79
DOI: 10.15439/2021F117
ISSN 2300-5963 ACSIS, Vol. 25

Stereotype-aware collaborative filtering

Gabriel Frisch, Jean-Benoist Leger, Yves Grandvalet
Université de Technologie de Compiègne
Heudiasyc Laboratory, UMR UTC/CNRS 7253

***Abstract*—In collaborative filtering, recommendations are made using user feedback on a few products. In this paper, we show that even if sensitive attributes are not used to fit the models, a disparate impact may nevertheless affect recommendations. We propose a definition of fairness for the recommender system that expresses that the ranking of items should be independent of sensitive attribute. We design a co-clustering of users and items that processes exogenous sensitive attributes to remove their influence to return fair recommendations. We prove that our model ensures approximately fair recommendations provided that the classification of users approximately respects statistical parity.**

I. Introduction

IN SIMPLE terms, fairness is often loosely defined as the quality of treating people equally, with impartiality and rightfulness. Although imprecise, this definition stipulates that equal treatment refers to certain sensitive attributes shared by groups of people, such as gender, age, ethnicity, socio-economic group, etc. In recent years, intensive research has highlighted the lack of fairness in decisions made by machine learning algorithms [6].

There are several stakeholders in a recommendation scenario. In the terminology of Burke *et al.* [8], we target consumer-fairness, where the objective is to provide the same treatment to users of the recommender system, regardless of their sensitive attribute. We target recommender systems relying on collaborative filtering, which aims at building recommendations from the history of user ratings. These observed ratings are the basis for making automatic predictions about non-rated items, under the assumption that users can be clustered according to their past opinion behavior. Sensitive attributes are not used to fit the models, but some disparate impacts may nevertheless exist, possibly due to some societal or cultural effects that bias the sampling of data [11]. In situations where the sensitive attribute can be collected, it therefore seems preferable to design algorithms that process sensitive attributes to remove their influence, rather than simply ignore them.

Many proposals have already been made on how fairness should be formally defined in collaborative filtering [12, 31]. One common approach is the recommendation independence [23], that requires the unconditional statistical independence between recommendations and a specified sensitive attribute. This equal treatment does not ensure equal impact (also called “equal opportunity”), which argues for equal recommendation quality between sensitive groups. Although some works [34] have argued that statistical parity may be overly restrictive, resulting in a poor quality of recommendations, we use here this definition to propose a fair collaborative filtering algorithm.

In this paper, we aim at producing fair recommendations using a co-clustering of users and items that respects statistical parity of users with respect to some sensitive attributes. For this purpose, we introduce a co-clustering model based on the Latent Block Model (LBM) that relies on an ordinal regression model that takes as inputs the sensitive attributes. We demonstrate that our model ensures approximately fair recommendations provided that the clustering of users approximately respects statistical parity. Finally, we conduct experiments on a real-world dataset to show that the proposed approach can help alleviate unfairness.

Related works

Several recent works have raised the issue of fairness in recommender systems. Kamishima *et al.* [23] have proposed methods for improving fairness, formalized as the independence of the predicted ratings with the sensitive attribute. Their methods are based on matrix factorization regularized by criteria that favor independence by controlling the moments of the distributions of rating among sensitive groups. Using the same definition of fairness, Zhu *et al.* [35] proposed a tensor method that isolates sensitive attributes in sub-dimensions of the latent factor matrix. Unlike many other methods, this solution is capable of handling multiple and non-binary sensitive attributes. Yao and Huang [34] proposed four new metrics that deal with different types of unfairness and used them as penalty functions in augmented matrix factorization objectives.

All of the above methods are based on the fairness of predicted ratings, but an approximate fairness of ratings may not entail an approximate fairness of the recommender system that provides users with a short list of relevant items. With this in mind, Beutel *et al.* [4] provided new metrics based on pairwise comparisons and proposed a novel pairwise regularization approach to improve the fairness of the recommender system during training. Finally, further from recommender systems but still related to the model we use, the notion of statistical parity is often considered for fairness in clustering methods [1, 14, 3].

IEEE Catalog Number: CFP2185N-ART ©2021, PTI

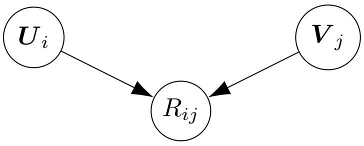


Fig. 1. Graphical view of the Latent Block Model. Entries R_{ij} of the data matrix are independently generated according to the group membership $\boldsymbol{U}_i$ of row i and the group membership $\boldsymbol{V}_j$ of column j.

II. Model

The data used to build recommender systems can be aggregated in a matrix where rows are users, columns are items and entries the feedbacks. The model we propose is based on the Latent Block Model that considers a data matrix to group users and items based on their opinions.

A. The Latent block models

The Latent block models (LBM), also known as bipartite stochastic block models and introduced in [15], are generative probabilistic models enabling to cluster jointly the rows and the columns of a data matrix denoted $\boldsymbol{R}$. These co-clustering models assume a homogeneous block structure of the whole data matrix. This structure is unveiled by the reordering of rows and columns according to their respective cluster index; for k_1 row clusters and k_2 column clusters, the reordering reveals $k_1 \times k_2$ homogeneous blocks in the data matrix being possibly binary [15] categorical [24], or quantitative [27, 16].

The partitions of rows and columns are governed by the latent variables $\boldsymbol{U}$ and $\boldsymbol{V}$, $\boldsymbol{U}$ being the $n_1 \times k_1$ indicator matrix of row classes, and $\boldsymbol{V}$ being the $n_2 \times k_2$ indicator matrix of the column classes. The class indicator of row i is denoted $\boldsymbol{U}_i$, and similarly, the class indicator of column j is denoted $\boldsymbol{V}_j$. The LBM makes several assumptions on the dependency and on the form of the distributions:

- The latent group memberships of rows and columns are assumed to be mutually independent and identically distributed, with respectively multinomial distributions $\mathcal{M}(1;\boldsymbol{\alpha})$ and $\mathcal{M}(1;\boldsymbol{\beta})$, where $\boldsymbol{\alpha} = (\alpha_1,\ldots,\alpha_{k_1})$ and $\boldsymbol{\beta} = (\beta_1,\ldots,\beta_{k_2})$ are the mixing proportions of rows and columns:
$$p(\boldsymbol{U},\boldsymbol{V}) = p(\boldsymbol{U})\,p(\boldsymbol{V}) = \prod_i p(\boldsymbol{U}_i;\boldsymbol{\alpha})\prod_j p(\boldsymbol{V}_j;\boldsymbol{\beta}) \ .$$
- Conditionally to rows and columns assignments $(\boldsymbol{U},\boldsymbol{V})$, the entries of the data matrix $\boldsymbol{R}$ are independent and identically distributed:
$$p(\,\boldsymbol{R}|\boldsymbol{U},\boldsymbol{V};\boldsymbol{\theta}\,) = \prod_{ij} p(\,R_{ij}|\boldsymbol{U}_i,\boldsymbol{V}_j\,) \ ,$$
$$p(\,R_{ij}|U_{iq}V_{jl}=1\,) = \phi_{ql}(R_{ij}) \ , \qquad (1)$$
with $\phi_{ql}(R_{ij})$ the density of the conditional distribution of R_{ij} depending on the group memberships of row i and column j.

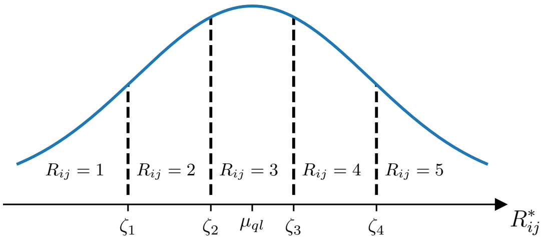


Fig. 2. The conditional density function of R^*_{ij} and its relationship to R_{ij}. Fixed thresholds ζ_k, defines the discretization of R^*_{ij}.

B. Model proposed

The user feedback used for collaborative filtering can be implicit (history, browsing history, clicks...) or explicit. In the case of explicit evaluation data, users most often express their interest in items using a discrete rating scale. This rating scale suppose an order between levels, for example from 1 to 5 expressing the worst opinion to the best one. Models handling this type of data can assume that these scales are a discretization of the opinion of a user that may be better handled by a continuous variable. The method we propose to model ratings is based on a statistical co-clustering using ordered probit regression to model ordinal responses. Covariates encoding a sensitive user attribute can easily be included in the probit regression framework.

1) Ordered probit in Latent Block Model: The ordered probit model [10] assumes the existence of a continuous, Gaussian distributed latent random variable, denoted $\boldsymbol{R^*}$. In a collaborative filtering context, this latent variable represents the underlying value, assumed to be continuous, assigned to an item by the user. The assumption of a single underlying continuous variable leading to ordinal ratings may be appropriate when ratings are not the result of a sequential process [9]. The discrete observed ratings $\boldsymbol{R}$ are the result of the partition of the continuous space of $\boldsymbol{R^*}$ by a set of thresholds $\boldsymbol{\zeta}$ such that: $R_{ij} = 1$ if $-\infty < R^*_{ij} < \zeta_1$, $R_{ij} = 2$ if $\zeta_1 < R^*_{ij} < \zeta_2$, …, $R_{ij} = K$ if $\zeta_{K-1} < R^*_{ij} < +\infty$ (see Figure 2).

We use the ordered probit model within a Latent Block Model (see Section II-A), assuming that conditionally to row and column group assignments, the entries of $\boldsymbol{R^*}$ are independent and identically distributed with Gaussian distribution:
$$p(\,R^*_{ij}|U_{iq}V_{jl}=1;\mu_{ql},\sigma\,) = \phi\big(R^*_{ij};\mu_{ql},\sigma^2\big) \ , \qquad (2)$$
with $\phi\big(\cdot;\mu_{ql},\sigma^2\big)$ the probability density function of the Gaussian distribution with mean $\mu_{ql} \in \mathbb{R}$ and variance $\sigma^2\mathbb{R}^*_+$. The conditional probability that a user i gives to the item j the rating with value k is then:
$$\begin{aligned} &p(\,R_{ij}=k|U_{iq}V_{jl}=1;\mu_{ql}\,) \\ &\quad = p(\,\zeta_{k-1} < R^*_{ij} < \zeta_k\big|U_{iq}V_{jl}=1;\mu_{ql}\,) \end{aligned}$$

$$= \Phi\big(\zeta_k; \mu_{ql}, \sigma^2\big) - \Phi\big(\zeta_{k-1}; \mu_{ql}, \sigma^2\big) \ ,$$

with $\Phi\big(\cdot; \mu_{ql}, \sigma^2\big)$ being the normal cumulative distribution function. To ensure model identifiability, the thresholds $\boldsymbol{\zeta}$ are fixed to equidistant predefined values.

2) Individual row and column effects: The Latent Block Model is well suited to collaborative filtering, in that it searches for users and items that share the same opinion patterns. However, a model that assumes that users in a given cluster share exactly the same opinion patterns is very restrictive. Instead, we assume here that opinions may be slightly different within a cluster, using a richer model than Equation (2) for the conditional distribution of R^*_{ij}. In addition to the cluster effect μ_{ql} derived solely from the group memberships of users and items, one deviation is induced by the user i and another by the item j :

$$p(\,R^*_{ij} \big| U_{iq}V_{jl} = 1, A_i, B_j; \mu_{ql}\,) = \phi\big(R^*_{ij}; \mu_{ql} + A_i + B_j, \sigma^2\big) \ , \tag{3}$$

with latent variables $\boldsymbol{A}$ and $\boldsymbol{B}$ independently and identically distributed with:

$$A_i \overset{\text{iid}}{\sim} \mathcal{N}\big(0, \sigma_A^2\big), \qquad \sigma_A^2 \in \mathbb{R}_+^*$$
$$B_i \overset{\text{iid}}{\sim} \mathcal{N}\big(0, \sigma_B^2\big), \qquad \sigma_B^2 \in \mathbb{R}_+^*$$

These two variables encode different rating patterns for users and items such as systematic over- or under-rating relative to the user or item populations.

3) Sensitive attribute: We assume that, in addition to the matrix of ratings, we have access to a sensitive attribute s_i, describing here a binary feature of user i that should not intervene in the recommendation of items (more general sensitive attributes are considered in Appendix VI-D). We introduce a latent variable C_j for each object j assuming that they interact with different strengths with the sensitive attribute. This interaction between the object j and the sensitive attribute s_i is added to the conditional distribution of R^*_{ij} (Equation 3):

$$p(\,R^*_{ij} \big| U_{iq}V_{jl} = 1, A_i, B_j, s_i, C_j; \mu_{ql}\,)$$
$$= \phi\big(R^*_{ij}; \mu_{ql} + A_i + B_j + s_iC_j, \sigma^2\big) \ ,$$

with

$$C_j \overset{\text{iid}}{\sim} \mathcal{N}\big(0, \sigma_C^2\big), \ \ \sigma_G^2 \in \mathbb{R}_+^* \ .$$

This model explains the ratings by $\mu_{ql} + A_i + B_j + s_iC_j$ and σ^2; the co-clustering is driven by μ_{ql}, and provided the effects of the sensitive attribute are well captured by s_iC_j, we expect the co-clustering to be independent of the sensitive attribute, which ensures fair recommendations as shown in Section III-B. A summary of the model we propose is presented in Figure 3.

4) Modelling missingness: The datasets extracted from recommender systems are usually extremely sparse, with a high proportion of missing ratings, that is, ratings that were not provided by the users. The model we proposed so far does not accommodate missing observations, and suppose a fully observed data matrix $\boldsymbol{R}$.

The study of missing data identifies three main type of missingness [33]: Missing Completely At Random (MCAR) and Missing At Random (MAR) referring to the mechanisms in which the probability of being missing does not depend on the variable of interest (here $\boldsymbol{R^*}$); and finally Missing Not At Random (NMAR) referring to the mechanisms in which the probability of being missing depends on the actual value of the missing data. A common implicit assumption in collaborative filtering is that ratings are MAR or MCAR: the presence/absence of ratings is assumed to convey no information whatsoever about the value of these ratings. For simplicity of statistical modelling we take the same assumption, although previous studies [28, 29] have shown a potential dependence between the presence of ratings and the underlying opinion. We introduce a simple Bernoulli missingness model generating $\boldsymbol{M} \in \{0,1\}^{n_1 \times n_2}$, a mask matrix where each entry M_{ij} is one with probability p and indicates whether the rating is observed: $M_{ij} = 1$ if R_{ij} is observed and 0 otherwise. Given the complete data matrix $\boldsymbol{R^*}$ and the mask matrix $\boldsymbol{M}$, the elements of the observed ratings $\boldsymbol{R}$ are generated as follows:

$$\big(R_{ij} \big| R^*_{ij}, M_{ij}\,\big) = \begin{cases} \sum_{k=1}^{K} k \ \mathbb{1}_{]\zeta_{k-1};\zeta_k]}(R^*_{ij}) & \text{if} \quad M_{ij} = 1 \\ \text{NA} & \text{if} \quad M_{ij} = 0 \end{cases}$$

Any generative model under a MCAR or MAR process can be fitted separately from the missingness model as the overall likelihood can be factorized between the observed and non observed data. Under such assumptions, we show in AppendixVI-A that ignoring non-observed ratings results in a proper fitting.

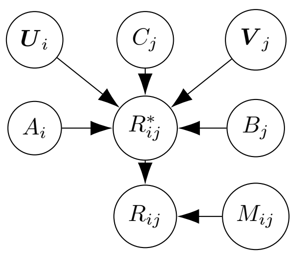
III. Inference and fair recommendations

A. A stochastic batch gradient descent of the variational criterion

The log-likelihood of the model is not tractable as it involves a sum that is combinatorially too large [7]. We resort to a variational inference procedure [20] that introduces q_γ, a restricted parametric inference distribution defined on the latent variables of the model, to optimize the following lower bound on the log-likelihood:

$$\mathcal{J}(\gamma, \theta) = \log p(\boldsymbol{R}; \theta) - \text{KL}\left(q_\gamma \,\|p(L|\boldsymbol{R})\,\right)$$

where KL stands for the Kullback-Leibler divergence, $\mathcal{H}$ for the differential entropy, $\theta = (\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\mu}, \sigma^2, \sigma_A^2, \sigma_B^2, \sigma_C^2, p)$ is the concatenation of the model parameters, and $L = (\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$ is the concatenation of the latent variables.



$$\begin{aligned}
\boldsymbol{U}_i &\overset{\text{iid}}{\sim} \mathcal{M}(1;\boldsymbol{\alpha}), & \boldsymbol{\alpha} &\in \mathbf{S}_{k_1-1}\\
\boldsymbol{V}_j &\overset{\text{iid}}{\sim} \mathcal{M}(1;\boldsymbol{\beta}), & \boldsymbol{\beta} &\in \mathbf{S}_{k_2-1}\\
A_i &\overset{\text{iid}}{\sim} \mathcal{N}\big(0,\sigma_A^2\big), & \sigma_A^2 &\in \mathbb{R}_+^*\\
B_j &\overset{\text{iid}}{\sim} \mathcal{N}\big(0,\sigma_B^2\big), & \sigma_B^2 &\in \mathbb{R}_+^*\\
C_j &\overset{\text{iid}}{\sim} \mathcal{N}\big(0,\sigma_C^2\big), & \sigma_C^2 &\in \mathbb{R}_+^*
\end{aligned}$$

$$\big(R^*_{ij}\big|U_{iq}=1,V_{jl}=1,A_i,B_j,C_j\big) \overset{\text{ind}}{\sim} \mathcal{N}\big(\mu_{ql}+A_i+B_j+s_iC_j,\sigma^2\big)$$

$$\big(R_{ij}\big|R^*_{ij},M_{ij}\big) = \begin{cases} \sum_{k=1}^{K} k\,\mathbb{1}_{]\zeta_{k-1};\zeta_k]}(R^*_{ij}) & \text{if} \quad M_{ij}=1\\ \text{NA} & \text{if} \quad M_{ij}=0 \end{cases}$$

with $M_{ij} \overset{\text{iid}}{\sim} \mathcal{B}(p), \qquad p \in [0,1]$

and $\zeta_0 = -\infty < \zeta_1 < \ldots < \zeta_{K-1} < \zeta_K = \infty$,

fixed thresholds

Fig. 3. Graphical view and summary of the ordered probit Latent Block Model with protected attribute $\boldsymbol{s}$. The discrete observed data R_{ij} is generated by the underlying continuous data R^*_{ij} and the mask entry M_{ij}.

The variational distribution q_γ is chosen so that the computation of the criterion becomes easier:

$$\begin{aligned}
&\forall i, \quad \boldsymbol{U}_i|\boldsymbol{R} \underset{q_\gamma}{\sim} \mathcal{M}\Big(1;\boldsymbol{\tau}_i^{(U)}\Big) && \forall j, \quad \boldsymbol{V}_j|\boldsymbol{R} \underset{q_\gamma}{\sim} \mathcal{M}\Big(1;\boldsymbol{\tau}_j^{(V)}\Big)\\
&\forall i, \quad A_i|\boldsymbol{R} \underset{q_\gamma}{\sim} \mathcal{N}\Big(\nu_i^{(A)},\rho_i^{(A)}\Big) && \forall j, \quad B_j|\boldsymbol{R} \underset{q_\gamma}{\sim} \mathcal{N}\Big(\nu_j^{(B)},\rho_j^{(B)}\Big)\\
& && \forall j, \quad C_j|\boldsymbol{R} \underset{q_\gamma}{\sim} \mathcal{N}\Big(\nu_j^{(C)},\rho_j^{(C)}\Big)
\end{aligned}$$

We also enforce the conditional independence of the latent variables, leading to the following fully factorized form:

$$\begin{aligned}
q_\gamma = \textstyle\prod_{i=1}^{n_1} \mathcal{M}\Big(1;\boldsymbol{\tau}_i^{(U)}\Big) \times \prod_{j=1}^{n_2} \mathcal{M}\Big(1;\boldsymbol{\tau}_j^{(V)}\Big) \qquad (4)\\
\times \textstyle\prod_{i=1}^{n_1} \mathcal{N}\Big(\nu_i^{(A)},\rho_i^{(A)}\Big) \times \prod_{j=1}^{n_2} \mathcal{N}\Big(\nu_j^{(B)},\rho_j^{(B)}\Big)\\
\times \textstyle\prod_{j=1}^{n_2} \mathcal{N}\Big(\nu_j^{(C)},\rho_j^{(C)}\Big) \quad ,
\end{aligned}$$

where γ denotes the concatenation of all parameters of the variational distribution[1]. This conditional independence of the latent variables to $\boldsymbol{R}$ simplifies the criterion $\mathcal{J}(\gamma,\theta)$ to:

$$\mathcal{J}(\gamma,\theta) = \mathbb{E}_{q_\gamma}[\log p(\boldsymbol{R}|L)] - \mathrm{KL}\left(q_\gamma \,\|p(L;\theta)\right) \ . \qquad (5)$$

As explained in Section II-B4, the optimization criterion relies only on the non-missing entries of $\boldsymbol{R}$ because the data is assumed to be missing at random. The full expansion of the criterion is given in Appendix VI-A.

We resort to a batch stochastic optimization to maximize the variational criterion using noisy estimates of its gradient [30]. Samples are drawn from the variational distribution (Equation 4) to estimate a noisy but unbiased gradient of the expectation of the conditional log-distribution of $\boldsymbol{R}$ (first term of Equation 5), which we then use to update our parameters as follows:

$$(\gamma,\theta)^{(t+1)} = (\gamma,\theta)^{(t)} + \eta \cdot \nabla_{(\gamma,\theta)}\mathcal{J}\big(\boldsymbol{R}_{(i:i+n),(j:j+n)};\gamma,\theta\big) \ ,$$

where n is the batch size and η is the adaptive learning rate based on the past gradients that were computed (Adam optimizer [25]).

Using a stochastic gradient algorithm instead of the usual EM algorithm alleviates the well-known initialization problems of the Latent Block Model, which result in unsatisfactory local maxima [5, 2]. However, it requires the use of differentiable functions to back-propagate gradients through the automatic differentiation graph.For this purpose, the multinomial distributions are replaced by a differentiable Gumbel-Softmax distribution [21].

B. Fair recommendations

This section describes a theoretical result establishing a guarantee on the fairness of recommendations. This guarantee is subject to an assumption about the parity of the clustering of users that can be tested in practice, and that holds true for the experiments reported in Section IV and Appendix VI-D. We develop here the case of a binary sensitive attribute to simplify the exposition. The result is more general and applies to any discrete sensitive attribute. It is proven in this general sense in Appendix VI-C.

Recommendations are partial orders between items. In collaborative filtering, the usual approach to producing recommendations is to estimate a relevance score for each item, which is then used to define a total order through numerical comparisons. With the parameters obtained by variational inference, we define the relevance score of item j for user i as:

$$\hat{R}_{ij} = \boldsymbol{\tau}_i^{(U)}\hat{\boldsymbol{\mu}}{\boldsymbol{\tau}_j^{(V)}}^T + \nu_i^{(A)} + \nu_j^{(B)} \ . \qquad (6)$$

This relevance score is computed from the maxima *a posteriori* of the latent variables encoding the user and item group memberships $(\boldsymbol{\tau}_i^{(U)},\boldsymbol{\tau}_j^{(V)})$, that is, the trend related to the co-cluster to which (i,j) belongs, and the global effects related to user i and item j. It does not use the user's sensitive attribute s_i which is considered here as a nuisance parameter, properly taken into account during inference and then ignored when predicting a relevance score. It then becomes possible to compare items fairly with respect to the sensitive attribute.

[1] $\gamma = (\boldsymbol{\tau}^{(U)},\boldsymbol{\tau}^{(V)},\boldsymbol{\nu^{(A)}},\boldsymbol{\rho^{(A)}},\boldsymbol{\nu^{(B)}},\boldsymbol{\rho^{(B)}},\boldsymbol{\nu^{(C)}},\boldsymbol{\rho^{(C)}})$

Definition III.1 (Fair comparison of items)**.** Given user i and any two items j and j', the comparison of items j and j' is said to be fair if it is freed from the evaluation bias regarding the sensitive attribute s: item j is fairly preferred to item j' if $\hat{R}_{ij} > \hat{R}_{ij'}$, that is:

$$\boldsymbol{\tau}_i^{(U)} \hat{\boldsymbol{\mu}} \boldsymbol{\tau}_j^{(V)^T} + \nu_i^{(A)} + \nu_j^{(B)} > \boldsymbol{\tau}_i^{(U)} \hat{\boldsymbol{\mu}} \boldsymbol{\tau}_{j'}^{(V)^T} + \nu_i^{(A)} + \nu_{j'}^{(B)} \ .$$

The modelling of the observed data $\boldsymbol{R}$ incorporates the term $\nu_j^{(C)} s_i$, interpreted here as a spurious opinion bias related to the sensitive attribute. While it is important to ignore this term for a fair comparison of items, its inclusion into the model is important to allow the construction of clusters that are not affected by this spurious effect. These clusters can then be expected to be representative of all subpopulations defined by their sensitive attribute value, and thus to respect the statistical parity of users.

Definition III.2 (Clustering ε-parity, binary sensitive attribute)**.** The clustering of users is said to respect ε-parity with respect to attribute s iff:

$$\forall q, \left| \frac{\#\left\{i | s_i = 1 \wedge u_{iq} = 1\right\}}{\#\left\{i | s_i = 1\right\}} - \frac{\#\left\{i | s_i = -1 \wedge u_{iq} = 1\right\}}{\#\left\{i | s_i = -1\right\}} \right| \leq \varepsilon \ , \tag{7}$$

where $\varepsilon \in \mathbb{R}_+$ measures the gap to exact parity, u_{iq} is the (hard) membership of user i to cluster q,and $\#\left\{i|\Omega\right\}$ is the number of users defined by the cardinality of the set Ω.

In essence, clustering ε-parity requires that subpopulations of users defined by identical sensitive attributes be represented approximately equally in each user group. For the Latent Block Model, the hard membership u_{iq} of Definition III.2 is given by the maximum *a posteriori* of the latent variable $\tau_{iq}^{(U)}$.

Our theoretical guarantee ensures that this approximate statistical parity in clusters is sufficient to get approximately fair recommendations from our model:

Definition III.3 (ε-fair recommendation, binary sensitive attribute)**.** A recommender system is said to be ε-fair with respect to attribute s if for any two items j and j':

$$\left| \frac{\#\left\{i | s_i = 1 \wedge (\hat{R}_{ij} > \hat{R}_{ij'})\right\}}{\#\left\{i | s_i = 1\right\}} - \frac{\#\left\{i | s_i = -1 \wedge (\hat{R}_{ij} > \hat{R}_{ij'})\right\}}{\#\left\{i | s_i = -1\right\}} \right| \leq \varepsilon \ , \tag{8}$$

where $\varepsilon \in \mathbb{R}_+$ measures the gap to exact fairness

In essence, an ε-fair recommender system ensures that, for any two items, the proportion of users with the same preference is approximately identical in all the subpopulations of users defined by identical sensitive attributes.

Theorem III.1 (Fair recommendation from clustering parity)**.** If the clustering of users in k_1 groups respects ε-parity (Definition III.2 or Definition VI.1) then the recommender system relying on the relevance score defined in Equation (6) is $(k_1\varepsilon)$-fair (Definition III.3 or Definition VI.2).

Proof: see Appendix VI-C.

IV. Experiment on MovieLens dataset

The final goal of a recommender system is to provide users with a shortlist of items that they might most enjoy. We choose here to directly assess the quality of the ranking rather than using proxy measures, such as root mean square error on ratings, that ignore relative rankings.

To measure the ranking performance of algorithms, we use the Normalized Discounted Cumulative Gain [22] (NDCG) that measures ranking quality by a penalized sum of the relevance scores of the ranking results:

$$NDCG@k = \frac{DCG@k}{IDCG@k} \text{ with } DCG@k = \sum_{i=1}^{k} \frac{rel_i}{\log(i+1)} \ ,$$

rel_i, the relevance of the results at each rank i before k and $IDCG@k$ being the $DCG@k$ computed with a perfect ranking.

We use the MovieLens 1M dataset [18] that contains one million ratings given by 6,040 users to 3,900 movies scaling from 1 to 5 (from least liked to most liked). The dataset also contains additional information about users: gender (binary), age category (seven levels) or occupation. We give here some experimental results where gender is the sensitive attribute, and additional results, in particular with age considered as the sensitive attribute, can be found in Appendix VI-D.

A. Experimental Protocol

We estimate the average performances by predicting preferences on ratings that are concealed during training. These concealed ratings form our test set, with 20 ratings per user, which is about 10% of the available data. This process is repeated 5 times, with independent random draws, to produce stable average performances.

We compare our model (referred to as Parity LBM) with the baseline LBM that does not use the sensitive variable in the modelling (referred to as Standard LBM). We expect the latter model to create groups of users that do not respect clustering parity and to generate unfair recommendations. We also compare to another co-clustering algorithm, weighted Bregman co-clustering [13] (referred to as Bregman co-clust) to compare the statistical parity of user groups inferred from another baseline. Finally, we compare with Singular Value Decomposition (SVD), a method popularized during the Netflix challenge [26] that still remains state of the art in collaborative filtering [32]. All these baselines are implemented in the Python module `Surprise` [19].

The number of clusters in co-clustering and the number of factors in matrix factorization are both arbitrarily set to fifteen. Another comparison with more clusters, provided in Appendix VI-D, produces qualitatively similar results.

We repeat the learning process 25 times from different random initializations to mitigate the initialization dependence that affects all optimization procedures. We select the best solution based on the optimization criteria, that

TABLE I

MEASURES OF STATISTICAL GENDER PARITY AMONG USER CLUSTERS. THE NUMBER OF USER GROUPS IS $k_1 = 15$. THE χ^2 STATISTIC (WITH 14 DEGREES OF FREEDOM) IS AVERAGED OVER THE FIVE REPLICATES OF THE EXPERIMENT. A HIGH VALUE OF THE χ^2 STATISTIC (OR A LOW P-VALUE) LEADS TO THE REJECTION OF THE CLUSTERING PARITY HYPOTHESIS.

Model	Parity LBM	Standard LBM	Bregman co-clust
χ^2 statistic	18.0	44.4	187
p-value	0.20	$5.1 \cdot 10^{-5}$	$< 10^{-15}$.

is, the one with the highest likelihood for the LBM models and the lowest training reconstruction error for the other baselines.

B. Results and Discussion

1) Gender as sensitive attribute: User gender (binary in this dataset) is used as the sensitive attribute s_i. In the dataset, 27% of users self-identified as females, this proportion must be met in each group to respect clustering parity. To measure the dependence between gender and user group memberships, we compute the χ^2 statistic constructed from the contingency table of males and females counts in each group. Table I reports the p-value for testing the independence between groups and genders, with an asymptotical test. We recall that, under the null hypothesis of independence, the test statistic with k degrees of freedom has mean k and variance $2k$. The results show that the methods that do not consider the sensitive variable in the modelling create groups that are dependent on gender. In contrast, our Parity-LBM model is consistent with the clustering parity hypothesis: the gender representation in groups is representative of the gender distribution in the overall dataset.

The fairness of recommendations resulting from this clustering parity is ascertained by computing the gap ε from exactly fair recommendations, as defined in Definition III.3. Figure 4 displays these gaps, with lower values indicating a fairer recommendation; our model provides a significantly fairer recommendation compared to the standard Latent Block Model, which is itself much fairer than the two other baselines. The order observed in Table I is followed.

Figure 5 depicts the ranking performance of algorithms with the NDCG, averaged over all users, for a recommendation list of 10 items. SVD gets the best overall result, followed by the Latent Block Models that outperform Bregman co-clustering. The overall performances of our model and the standard LBM are not significantly different. Figure 5 also reports the average NDCG within each sensitive group. This performance measure shows that female users receive significantly less relevant recommendations than males with all algorithms. This measure of disparate impact on truly relevant recommendations is reminiscent of equalized odds [17] in the classification framework, in that it measures a disparity on positive outcomes. The performance gap between the sensitive groups is reduced by our parity LBM compared to the standard LBM. Although the difference is the smallest among all comparisons, our model does not eliminate disparate impact. As a cautionary note, although it is likely that the recommendations are less relevant to female users, under the assumption that the observed ratings are somewhat influenced by gender stereotypes, it is not possible to satisfactorily measure the performance of fair recommendations from the original rating matrix.

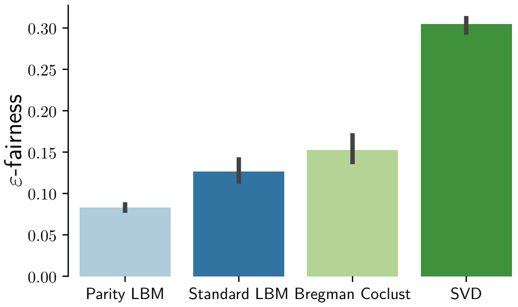


Fig. 4. Gaps ε for the ε-fair recommendations (see Definition III.3) provided by each model: a smaller ε-fairness indicates fairer recommendations.

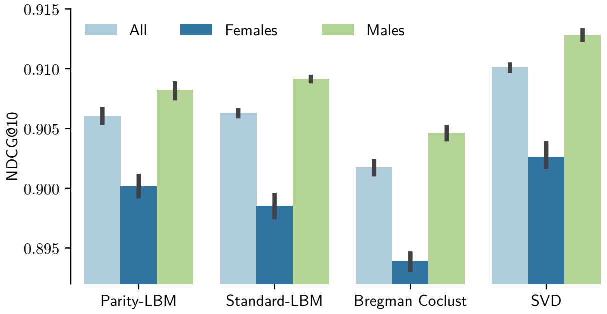


Fig. 5. Normalized Discounted Cumulative Gain estimated on MovieLens-1M (the higher the better)

Finally, we present some insights provided by our model on movies. We recall that the latent variable C_j, which is not used for fair prediction, captures the difference in opinion trends between female and male users on movie j. A high absolute value of C_j indicates a strongly gendered opinion for movie j. With our encoding of genres, negative C_j indicate a relative overrating by females and positive C_j indicate a relative overrating by males. We display the empirical cumulative distribution function (CDF) of C_j for movies conditionally on their genre (for some handpicked archetypal genres). The dominance of the CDF for a given genre expresses that, according to our model, female users have a higher opinion than male users for the movies belonging to that genre. Figure 6 shows the

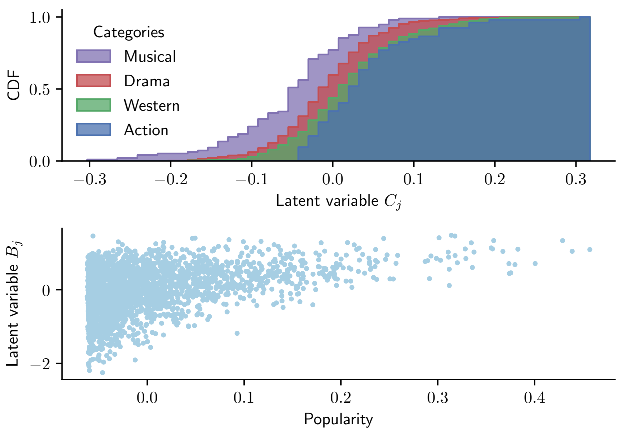


Fig. 6. Top: cumulative distribution function of latent variable C_j conditionally on the genre of the movie. A dominating CDF indicates a genre for which females' opinions are more positive than males'. Bottom: scatter plot of the movie latent variable B_j versus popularity (ratio of ratings). High positive values of B_j (resp. popularity) correspond to movies that are the most liked (resp. popular).

results, which reflect stereotypes that women are more likely than men to positively evaluate musical films and dramas, while men are similarly inclined toward westerns and action films. These stereotypes are incorporated into our model to fit actual ratings, but ignored to deliver fair recommendations. The lists of extreme movies based on extreme (positive and negative) values of C_j is given in Appendix VI-D1.

The latent variable B_j encodes the overall opinion trend about movie j. Two interesting observations can be made from the scatter plot of B_j versus movie popularity (see bottom of Figure 6). First, unpopular movies are also the least appreciated according to our model; this supports the hypothesis that ratings are generated by a MNAR (Missing Not At Random) process, where a missing rating can be considered as weak negative feedback, assuming that users primarily rate items they like. This missingness process must still be taken into account in our model. Second, it shows that the most liked movies (according to our model) are not necessarily the most popular (and will be recommended); the recommendations are not affected by popularity bias.

V. Conclusion

We proposed a new co-clustering method for fair recommendation. Our model combines the Gaussian Latent Block Model with an ordinal regression model. The sensitive attribute is adequately accounted for in the model, allowing the clustering of users to be unaffected by the effects of this attribute on ratings. This results in user clusters that approximately respect statistical parity. We base recommendation on a relevance score that ignores the sensitive attribute in order to compare items fairly. We provide theoretical guarantees ensuring approximately fair recommendations, for any known discrete sensitive attribute, provided that the clustering of users respects an approximate statistical parity that can be assessed in practice. Our analysis focuses on the fairness of preferences, as defined by the ranking of ratings, rather than on the predicted values themselves, which are less relevant for recommendation. Through experiments on real-world data, we show that our method significantly mitigates the unfairness of recommendations. Furthermore, the latent variables inferred by the model are also amenable to analyses that can help identify recommendation bias.

Our study supports that the absence of rating conveys some information that should be exploited. Previous works [28, 29] have already shown that the data used for collaborative filtering datasets can be strongly influenced by observational bias, which motivates dealing with missingness by a Missing Not At Random (MNAR) process. Societal biases may have a significant contribution to missingness, leading to an additional source of unfairness if missingness is not properly modeled. Studying fairness with MNAR processes is a highly relevant but extremely challenging direction for future research, as assessing the relevance of MNAR models in real situations requires data that are typically produced by online randomized experiments.

VI. Appendix

A. Computation of the variational log-likelihood criterion

The criterion we want to optimize is:

$$\mathcal{J}(q_\gamma, \theta) = \mathcal{H}(q_\gamma) + \mathbb{E}_{q_\gamma}\left[\mathcal{L}(\boldsymbol{R}, \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}; \theta)\right] \quad . \tag{9}$$

We chose to restrict the space of the variational distribution q_γ in order to get a fully factorized form:

$$\begin{aligned} q_\gamma = & \textstyle\prod_{i=1}^{n_1} \mathcal{M}\left(1; \tau_i^{(U)}\right) \times \prod_{j=1}^{n_2} \mathcal{M}\left(1; \tau_j^{(V)}\right) \\ & \times \textstyle\prod_{i=1}^{n_1} \mathcal{N}\left(\nu_i^{(A)}, \rho_i^{(A)}\right) \times \prod_{j=1}^{n_2} \mathcal{N}\left(\nu_j^{(B)}, \rho_j^{(B)}\right) \\ & \times \textstyle\prod_{j=1}^{n_2} \mathcal{N}\left(\nu_j^{(C)}, \rho_j^{(C)}\right) \end{aligned} \tag{10}$$

where γ denotes the parameters concatenation of the variational distribution[2] q_γ. The entropy is additive across independant variables so we get:

$$\begin{aligned} \mathcal{H}(q_\gamma) = & \mathcal{H}(q_\gamma(\boldsymbol{U})) + \mathcal{H}(q_\gamma(\boldsymbol{V})) \\ & + \mathcal{H}(q_\gamma(\boldsymbol{A})) + \mathcal{H}(q_\gamma(\boldsymbol{B})) + \mathcal{H}(q_\gamma(\boldsymbol{C})) \ , \end{aligned}$$

[2] $\gamma = (\boldsymbol{\tau}^{(U)}, \boldsymbol{\tau}^{(V)}, \boldsymbol{\nu}^{(\boldsymbol{A})}, \boldsymbol{\rho}^{(\boldsymbol{A})}, \boldsymbol{\nu}^{(\boldsymbol{B})}, \boldsymbol{\rho}^{(\boldsymbol{B})}, \boldsymbol{\nu}^{(\boldsymbol{C})}, \boldsymbol{\rho}^{(\boldsymbol{C})})$

with the following terms:

$$\begin{aligned}
\mathcal{H}(q_\gamma(\boldsymbol{U})) &= -\sum_{iq} \tau_{iq}^{(U)} \log \tau_{iq}^{(U)} \\
\mathcal{H}(q_\gamma(\boldsymbol{V})) &= -\sum_{jl} \tau_{jl}^{(U)} \log \tau_{jl}^{(V)} \\
\mathcal{H}(q_\gamma(\boldsymbol{A})) &= \frac{1}{2}\sum_{i} \log \rho_i^{(A)} + \frac{n_1}{2}(\log 2\pi + 1) \\
\mathcal{H}(q_\gamma(\boldsymbol{B})) &= \frac{1}{2}\sum_{j} \log \rho_j^{(B)} + \frac{n_2}{2}(\log 2\pi + 1) \\
\mathcal{H}(q_\gamma(\boldsymbol{C})) &= \frac{1}{2}\sum_{j} \log \rho_j^{(C)} + \frac{n_2}{2}(\log 2\pi + 1)
\end{aligned}$$

The independence of the latent variables allows to rewrite the expectation of the complete log-likelihood as:

$$\begin{aligned}
\mathbb{E}_{q_\gamma}[\mathcal{L}(\boldsymbol{R},\boldsymbol{U},\boldsymbol{V},\boldsymbol{A},\boldsymbol{B},\boldsymbol{C})] = \mathbb{E}_{q_\gamma}[\mathcal{L}(\boldsymbol{U})] &+ \mathbb{E}_{q_\gamma}[\mathcal{L}(\boldsymbol{V})] \\
&+ \mathbb{E}_{q_\gamma}[\mathcal{L}(\boldsymbol{A})] + \mathbb{E}_{q_\gamma}[\mathcal{L}(\boldsymbol{B})] \\
&+ \mathbb{E}_{q_\gamma}[\mathcal{L}(\boldsymbol{C})] \\
&+ \mathbb{E}_{q_\gamma}[\mathcal{L}(\,\boldsymbol{R}|\boldsymbol{U},\boldsymbol{V},\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}\,)] \ ,
\end{aligned}$$

with the following terms:

$$\begin{aligned}
\mathbb{E}_{q_\gamma}\mathcal{L}(\boldsymbol{U}) &= \mathbb{E}_{q_\gamma}\left[\sum_{iq} U_{iq}\log\alpha_q\right] = \sum_{iq}\tau_{iq}^{(U)}\log\alpha_q \\
\mathbb{E}_{q_\gamma}\mathcal{L}(\boldsymbol{V}) &= \mathbb{E}_{q_\gamma}\left[\sum_{jl} V_{jl}\log\beta_l\right] = \sum_{jl}\tau_{jl}^{(V)}\log\beta_l
\end{aligned}$$

$$\begin{aligned}
\mathbb{E}_{q_\gamma}\mathcal{L}(\boldsymbol{A}) &= -\frac{n_1}{2}\log 2\pi - \frac{n_1}{2}\log\sigma_A^2 - \frac{1}{2\sigma_A^2}\sum_i \mathbb{E}_{q_\gamma}A_i^2 \\
&= -\frac{n_1}{2}\log 2\pi - \frac{n_1}{2}\log\sigma_A^2 - \frac{1}{2\sigma_A^2}\sum_i\left(\left(\nu_i^{(A)}\right)^2 + \rho_i^{(A)}\right) \\
\mathbb{E}_{q_\gamma}\mathcal{L}(\boldsymbol{B}) &= -\frac{n_2}{2}\log 2\pi - \frac{n_2}{2}\log\sigma_B^2 - \frac{1}{2\sigma_B^2}\sum_i\left(\left(\nu_i^{(B)}\right)^2 + \rho_i^{(B)}\right) \\
\mathbb{E}_{q_\gamma}\mathcal{L}(\boldsymbol{C}) &= -\frac{n_2}{2}\log 2\pi - \frac{n_2}{2}\log\sigma_C^2 - \frac{1}{2\sigma_C^2}\sum_j\left(\left(\nu_j^{(C)}\right)^2 + \rho_j^{(C)}\right)
\end{aligned}$$

and as the entries of the data matrix $\boldsymbol{R}$ are independent and identically distributed:

$$\begin{aligned}
&\mathbb{E}_{q_\gamma}\mathcal{L}(\,\boldsymbol{R}|\boldsymbol{A},\boldsymbol{B},\boldsymbol{C},\boldsymbol{U},\boldsymbol{V}\,) = \\
&\mathbb{E}_{q_\gamma}\mathcal{L}\Big(\boldsymbol{R}^{(\mathrm{o})}\Big|\boldsymbol{A},\boldsymbol{B},\boldsymbol{C},\boldsymbol{U},\boldsymbol{V}\Big) + \mathcal{L}\Big(\boldsymbol{R}^{(\neg o)}\Big) \qquad (11)
\end{aligned}$$

where $\boldsymbol{R}^{(\mathrm{o})}$ denotes the set of observed ratings and $\boldsymbol{R}^{(\neg o)}$, the set of non-observed ratings, where $R_{ij} = \mathrm{NA}$. From Equation 11, it becomes clear that maximizing $\mathbb{E}_{q_\gamma}\mathcal{L}(\boldsymbol{R}^{(\neg o)})$ is not necessary to infer the model parameters used for prediction and therefore ignoring the non-observed data is correct. The expectation of the conditional log-likelihood (first term of right side of Equation 11) is numerically estimated by sampling from q_γ.

Stochastic gradient optimization To optimize the criterion with stochastic gradient descent, we express the variational log-likelihood criterion on a single rating:

$$\begin{aligned}
\mathcal{J}(R_{ij};q_\gamma,\theta) = \mathbb{E}_{q_\gamma}&\Big[\mathcal{L}\Big(\,R_{ij}^{(\mathrm{o})}\Big|\boldsymbol{U}_i,\boldsymbol{V}_j,A_i,B_j,C_j\,\Big)\Big] \\
&+ \frac{1}{n_2}\big(\mathcal{H}(q_\gamma(\boldsymbol{U}_i)) + \mathcal{H}(q_\gamma(A_i)) + \mathbb{E}_{q_\gamma}[\mathcal{L}(\boldsymbol{U}_i)] + \mathbb{E}_{q_\gamma}[\mathcal{L}(A_i)]\big) \\
&+ \frac{1}{n_2}\big(\mathcal{H}(q_\gamma(\boldsymbol{V}_j)) + \mathcal{H}(q_\gamma(B_j)) + \mathbb{E}_{q_\gamma}[\mathcal{L}(\boldsymbol{V}_j)] + \mathbb{E}_{q_\gamma}[\mathcal{L}(B_j)]\big) \\
&+ \frac{1}{n_2}\big(\mathcal{H}(q_\gamma(C_j)) + \mathbb{E}_{q_\gamma}[\mathcal{L}(C_j)]\big)
\end{aligned}$$

A batch of data, $\boldsymbol{R}_{(i:i+n),(j:j+n)}$, consists of a $(n\times n)$ submatrix randomly sampled from the original matrix $\boldsymbol{R}$.

B. Clustering ε-parity and ε-fair recommendation for arbitrary discrete sensitive attribute

Definition VI.1 (Clustering ε-parity, arbitrary discrete sensitive attribute)**.** The clustering of users is said to respect ε-parity with respect to the discrete attribute $s \in \mathcal{S}$ iff:

$$\begin{aligned}
&\forall (t,t') \in \mathcal{S}^2,\ \forall q, \\
&\left|\frac{\#\{i|s_i = t \wedge u_{iq} = 1\}}{\#\{i|s_i = t\}} - \frac{\#\{i|s_i = t' \wedge u_{iq} = 1\}}{\#\{i|s_i = t'\}}\right| \le \varepsilon\ ,
\end{aligned} \tag{12}$$

where $\varepsilon \in \mathbb{R}_+$ measures the gap to exact parity, u_{iq} is the (hard) membership of user i to cluster q,and $\#\{i|\Omega\}$ is the number of users defined by the cardinality of the set Ω.

Definition VI.2 (ε-fair recommendation, arbitrary discrete sensitive attribute)**.** A recommender system is said to be ε-fair with respect to the dicrete attribute $s \in \mathcal{S}$ if for any two items j and j': $\forall (t,t') \in \mathcal{S}^2$,

$$\left|\frac{\#\left\{i|s_i = t \wedge (\hat{R}_{ij} > \hat{R}_{ij'})\right\}}{\#\{i|s_i = t\}} - \frac{\#\left\{i|s_i = t' \wedge (\hat{R}_{ij} > \hat{R}_{ij'})\right\}}{\#\{i|s_i = t'\}}\right| \le \varepsilon\ , \tag{13}$$

where $\varepsilon \in \mathbb{R}_+$ measures the gap to exact fairness

C. Proof of Theorem III.1

Theorem VI.1 (Fair recommendation from clustering parity)**.** *If the clustering of users in k_1 groups respects ε-parity (Definition III.2 or Definition VI.1) then the recommender system relying on the relevance score defined in Equation (6) is $(k_1\varepsilon)$-fair (Definition III.3 or Definition VI.2).*

Proof. Suppose that $\boldsymbol{\tau}^{(U)}$, the maximum *a posteriori* of $\boldsymbol{U}$, is a binary matrix; $\boldsymbol{\tau}^{(U)}$ is thus a $n_1 \times k_1$ indicator matrix of row classes membership. Then, given user i, item

j is said to be preferred to item j' if $\hat{R}_{ij} > \hat{R}_{ij'}$, that is:

$$\begin{aligned}
\hat{R}_{ij} > \hat{R}_{ij'} &\iff \boldsymbol{\tau}_i^{(U)} \hat{\boldsymbol{\mu}} \boldsymbol{\tau}_j^{(V)^T} + \nu_i^{(A)} + \nu_j^{(B)} \\
&\qquad > \boldsymbol{\tau}_i^{(U)} \hat{\boldsymbol{\mu}} \boldsymbol{\tau}_{j'}^{(V)^T} + \nu_i^{(A)} + \nu_{j'}^{(B)} \\
&\iff \boldsymbol{\tau}_i^{(U)} \hat{\boldsymbol{\mu}} \left(\boldsymbol{\tau}_j^{(V)} - \boldsymbol{\tau}_{j'}^{(V)} \right)^T > \nu_{j'}^{(B)} - \nu_j^{(B)} \\
&\iff \boldsymbol{\tau}_i^{(U)} \boldsymbol{a} > b \\
&\iff \boldsymbol{a}_{d_i} > b \ , \qquad (14)
\end{aligned}$$

with $\boldsymbol{a} \in \mathbb{R}^{k_1}$ defined by $\boldsymbol{a} = \hat{\boldsymbol{\mu}} \left(\boldsymbol{\tau}_j^{(V)} - \boldsymbol{\tau}_{j'}^{(V)} \right)^T$, $b \in \mathbb{R}$ defined by $b = \nu_{j'}^{(B)} - \nu_j^{(B)}$ and $d_i \in \{1, \cdots, k_1\}$ being the group indicator of user i: $\tau_{i,d_i}^{(U)} = 1$.

Suppose ε-parity, from Definition VI.1 (Definition III.2 is a particular case of Definition VI.1), we have

$$\forall (t, t'), \qquad \forall q,$$
$$\left| \frac{\# \{i | s_i = t \wedge d_i = q\}}{\# \{i | s_i = t\}} - \frac{\# \{i | s_i = t' \wedge d_i = q\}}{\# \{i | s_i = t'\}} \right| \leq \varepsilon$$

therefore,

$$\forall (t, t'), \quad \forall q,$$
$$\left| \mathbb{1}_{\boldsymbol{a}_{d_i} > b} \frac{\# \{i | s_i = t \wedge d_i = q\}}{\# \{i | s_i = t\}} - \mathbb{1}_{\boldsymbol{a}_{d_i} > b} \frac{\# \{i | s_i = t' \wedge d_i = q\}}{\# \{i | s_i = t'\}} \right| \leq \varepsilon \mathbb{1}_{\boldsymbol{a}_{d_i} > b}$$

By summing over all groups, we get:

$$\forall (t, t'),$$
$$\sum_q \left| \frac{\mathbb{1}_{\boldsymbol{a}_{d_i} > b} \# \{i | s_i = t \wedge d_i = q\}}{\# \{i | s_i = t\}} - \frac{\mathbb{1}_{\boldsymbol{a}_{d_i} > b} \# \{i | s_i = t' \wedge d_i = q\}}{\# \{i | s_i = t'\}} \right|$$
$$\leq \varepsilon \sum_q \mathbb{1}_{\boldsymbol{a}_{d_i} > b}$$

and from the triangular inequality, $\forall (t, t')$:

$$\left| \frac{\sum_q \mathbb{1}_{\boldsymbol{a}_{d_i} > b} \# \{i | s_i = t \wedge d_i = q\}}{\# \{i | s_i = t\}} - \frac{\sum_q \mathbb{1}_{\boldsymbol{a}_{d_i} > b} \# \{i | s_i = t' \wedge d_i = q\}}{\# \{i | s_i = t'\}} \right|$$
$$\leq \varepsilon \sum_q \mathbb{1}_{\boldsymbol{a}_{d_i} > b}$$
$$\Leftrightarrow \left| \frac{\# \{i | s_i = t \wedge \boldsymbol{a}_{d_i} > b\}}{\# \{i | s_i = t\}} - \frac{\# \{i | s_i = t' \wedge \boldsymbol{a}_{d_i} > b\}}{\# \{i | s_i = t'\}} \right| \leq \varepsilon k_1$$

And, applying (14), the result is obtained:

$$\forall (t, t'),$$
$$\left| \frac{\# \left\{ i | s_i = t \wedge (\hat{R}_{ij} > \hat{R}_{ij'}) \right\}}{\# \{i | s_i = t\}} - \frac{\# \left\{ i | s_i = t' \wedge (\hat{R}_{ij} > \hat{R}_{ij'}) \right\}}{\# \{i | s_i = t'\}} \right| \leq \varepsilon k_1$$

□

D. Supplemental results for MovieLens 1M

1) Gender as sensitive attribute:

a) Supplemental analysis of the model: We list in Tables II and III the most extreme movies according to the inferred value of their latent variable C_j. Variable C_j encodes the difference in opinion between the sensitive groups, not the overall opinion. For example, a movie may well be liked by most people but liked even more by males. Table II lists movies for which females have a better opinion than males and Table III lists movies for which males have a better opinion than females.

TABLE II
LIST OF MOVIES WITH THE LARGEST GAP IN OPINION BETWEEN FEMALES AND MALES FOR WHICH FEMALES HAVE A BETTER OPINION THAN MALES

Title	Year	Genders	Cj
Dirty Dancing	1987	Musical|Romance	0.31
Rocky Horror Picture Show, The	1975	Comedy|Horror|Musical|Sci-Fi	0.26
Sound of Music, The	1965	Musical	0.24
Grease	1978	Comedy|Musical|Romance	0.23
Jumpin' Jack Flash	1986	Action|Comedy|Romance|Thriller	0.23
Gone with the Wind	1939	Drama|Romance|War	0.22
Newsies	1992	Children's|Musical	0.21
Strictly Ballroom	1992	Comedy|Romance	0.21
Steel Magnolias	1989	Drama	0.20
Sense and Sensibility	1995	Drama|Romance	0.20
Full Monty, The	1997	Comedy	0.19
Much Ado About Nothing	1993	Comedy|Romance	0.18
Thelma & Louise	1991	Action|Drama	0.18
Swing Kids	1993	Drama|War	0.17
Fried Green Tomatoes	1991	Drama	0.17
Ever After: A Cinderella Story	1998	Drama|Romance	0.17
Anastasia	1997	Animation|Children's|Musical	0.17
Little Women	1994	Drama	0.17
Color Purple, The	1985	Drama	0.17
To Wong Foo, Thanks for Everything!	1995	Comedy	0.17

TABLE III
LIST OF MOVIES WITH THE LARGEST GAP IN OPINION BETWEEN FEMALES AND MALES FOR WHICH MALES HAVE A BETTER OPINION THAN FEMALES

Title	Year	Genders	Cj
Good, The Bad and The Ugly, The	1966	Action|Western	-0.32
Animal House	1978	Comedy	-0.30
Caddyshack	1980	Comedy	-0.27
Dumb & Dumber	1994	Comedy	-0.27
Exorcist, The	1973	Horror	-0.24
Clockwork Orange, A	1971	Sci-Fi	-0.24
Patton	1970	Drama|War	-0.23
Godfather: Part II, The	1974	Action|Crime|Drama	-0.22
Reservoir Dogs	1992	Crime|Thriller	-0.22
Saving Private Ryan	1998	Action|Drama|War	-0.22
Airplane!	1980	Comedy	-0.21
Eyes Wide Shut	1999	Drama	-0.21
Aliens	1986	Action|Sci-Fi|Thriller|War	-0.21
Predator	1987	Action|Sci-Fi|Thriller	-0.20
Apocalypse Now	1979	Drama|War	-0.20
Unforgiven	1992	Western	-0.20
Evil Dead II (Dead By Dawn)	1987	Action|Adventure|Comedy|Horror	-0.20
Big Trouble in Little China	1986	Action|Comedy	-0.20
Godfather, The	1972	Action|Crime|Drama	-0.20

b) Higher number of groups: We did not optimize the hyper-parameters of the compared models. We present here additional experiments to illustrate that the conclusions of Section IV apply to different hyper-parameter settings. Using a substantially larger number of groups ($k_1 = 50$ user groups and $k_2 = 50$ item groups) or a larger dimension of latent factors for SVD (also 50), the statistical gender parity measures given in Table IV and the recommendation performance given in Figure 7 are qualitatively similar to the ones given in Table I and Figure 5.

2) Age as sensitive attribute: The age range of the users is indicated within the following intervals: 'Under 18','18-24', '25-34', '35-44', '45-49', '50-55' and '56+'.

User age is treated as sensitive: we introduce seven

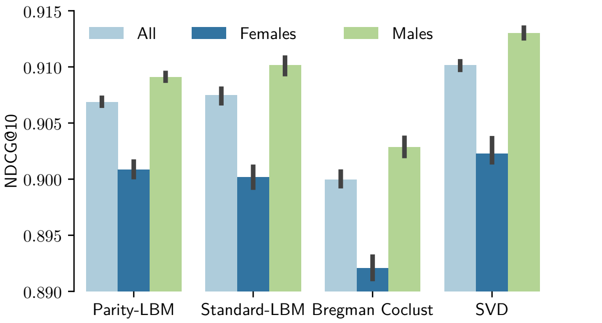


Fig. 7. Normalized Discounted Cumulative Gain estimated on MovieLens-1M with $k_1 = k_2 = 50$ groups for clustering methods and 50 factors for the SVD.

TABLE IV
MEASURES OF GENDER STATISTICAL PARITY. THE NUMBER OF USER GROUPS IS $k_1 = 50$. THE χ^2 STATISTIC (WITH 49 DEGREES OF FREEDOM) IS AVERAGED OVER THE FIVE REPLICATES OF THE EXPERIMENT. A HIGH VALUE OF THE χ^2 STATISTIC (OR A LOW P-VALUE) LEADS TO THE REJECTION OF THE STATISTICAL PARITY HYPOTHESIS.

Model	Parity LBM	Standard LBM	Bregman co-clust
χ^2 statistic	20	94	105
p-value	0.999	$1.1 \cdot 10^{-4}$	$5.8 \cdot 10^{-6}$

binary sensitive attributes s_i encoding for the seven categories of user age. We use a one-hot encoding of the seven categories of user age and introduce for the purpose seven binary sensitive attributes $s_i^1, \cdots, s_i^7$ and their item associated latent variables $C_j^1, \cdots, C_j^7$. We use the protocol described in Section IV with the exception that our Parity-LBM is initialized from estimates obtained with the Standard-LBM. Table V presents results of the χ^2 statistics constructed from the contingency table of user age counts in each group. The methods that do not consider the sensitive variable in the modelling create groups that are dependent on the age and assuming the statistical parity with our Parity-LBM model is reasonable.

Finally, we illustrate the interpretability of the estimates of the latent variables $C_j^1, \cdots, C_j^7$ related to movies. For each age category k, we select the thirty movies with the largest value of the latent variables C_j^k. These movies have the largest positive opinion bias for users in the given age category. Figure 8 displays a boxplot of the release years of these films for all user age categories. The greater variability in the distribution for older users means that they have a comparatively higher opinion of older movies than younger users. If user age were the sensitive attribute, the recommendations would not account for these differences.

TABLE V
MEASURES OF STATISTICAL PARITY WITH RESPECT TO AGE CATEGORY. THE NUMBER OF GROUP OF USERS IS $k_1 = 15$. A HIGH VALUE OF THE χ^2 STATISTIC (OR A LOW P-VALUE) LEADS TO THE REJECTION OF THE STATISTICAL PARITY HYPOTHESIS. THE χ^2 STATISTIC IS AVERAGED ON THE FIVE FOLDS OF THE CROSS-VALIDATION. DEGREES OF FREEDOM IS 14.

Model	Parity LBM	Standard LBM	Bregman co-clust
χ^2 statistic	99	144	577
p-value	0.12	$5.1 \cdot 10^{-5}$	$< 10^{-15}$

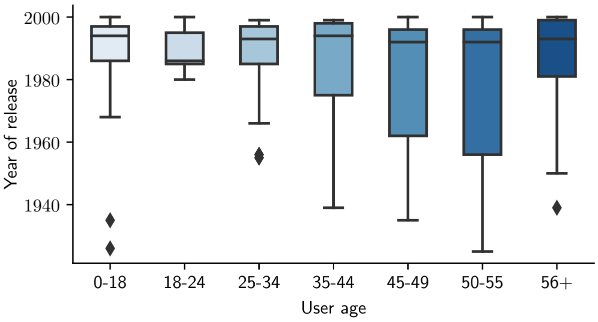


Fig. 8. Release years of the thirty most extreme movies according to the inferred positive value of the latent variables $C_j^1, \cdots, C_j^7$. Each latent variable C_j^k is matched with its corresponding user age category.

REFERENCES

[1] Mohsen Abbasi, Aditya Bhaskara, and Suresh Venkatasubramanian. Fair clustering via equitable group representations. In Madeleine Clare Elish, William Isaac, and Richard S. Zemel, editors, *ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, pages 504–514, 2021.

[2] Jean-Patrick Baudry and Gilles Celeux. EM for mixtures. *Statistics and Computing*, 25(4):713–726, 2015.

[3] Suman K. Bera, Deeparnab Chakrabarty, Nicolas J. Flores, and Maryam Negahbani. Fair algorithms for clustering, 2019.

[4] Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Li Wei, Yi Wu, Lukasz Heldt, Zhe Zhao, Lichan Hong, Ed H. Chi, and Cristos Goodrow. *Fairness in Recommendation Ranking through Pairwise Comparisons*, pages 2212—-2220. 2019.

[5] Christophe Biernacki, Gilles Celeux, and Gérard Govaert. Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models. *Computational Statistics & Data Analysis*, 41:561–575, 2003.

[6] Reuben Binns. Fairness in machine learning: Lessons from political philosophy. In Sorelle A. Friedler and Christo Wilson, editors, *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, volume 81 of *Proceedings of Machine Learning Research*, pages 149–159, New York, NY, USA, 23–24 Feb 2018. PMLR.

[7] Vincent Brault and Mahendra Mariadassou. Co-clustering through latent bloc model: A review. *Journal de la Société Française de Statistique*, 156(3):120–139, 2015.

[8] Robin Burke, Nasim Sonboli, and Aldo Ordonez-Gauger. Balanced neighborhoods for multi-sided fairness in recommendation. In *1st Conference on Fairness, Accountability and Transparency*, volume 81 of *PMLR*, pages 202–214, 2018.

[9] Paul-Christian Bürkner and Matti Vuorre. Ordinal regression models in psychology: A tutorial. *Advances in Methods and Practices in Psychological Science*, 2(1):77–101, 2019.

[10] Anne R. Daykin and Peter G. Moffatt. Analyzing ordered responses: A review of the ordered probit model. *Understanding Statistics*, 1(3):157–166, 2002.

[11] Thomas N. Daymonti and Paul J. Andrisani. Job preferences, college major, and the gender gap in earnings. *Journal of Human Resources*, 19(3):408–428, 1984.

[12] Pratik Gajane. On formalizing fairness in prediction with machine learning. *CoRR*, abs/1710.03184, 2017.

[13] Thomas George and Srujana Merugu. A scalable collaborative filtering framework based on co-clustering. In *Fifth IEEE International Conference on Data Mining (ICDM)*, 2005.

[14] Mehrdad Ghadiri, Samira Samadi, and Santosh Vempala. Socially fair k-means clustering. *arXiv preprint arXiv:2006.10085*, 2020.

[15] Gérard Govaert and Mohamed Nadif. Block clustering with Bernoulli mixture models: Comparison of different approaches. *Computational Statistics & Data Analysis*, 52(6):3233–3245, February 2008.

[16] Gérard Govaert and Mohamed Nadif. Latent block model for contingency table. *Communications in Statistics - Theory and Methods*, 39(3):416–425, 2010.

[17] Moritz Hardt, Eric Price, and Nathan Srebro. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems 29*, pages 3315–3323, 2016.

[18] F. Maxwell Harper and Joseph A. Konstan. The MovieLens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4), December 2015.

[19] Nicolas Hug. Surprise: A python library for recommender systems. *Journal of Open Source Software*, 5(52):2174, 2020.

[20] Tommi S. Jaakkola. Tutorial on variational approximation methods. In *Advanced Mean Field Methods: Theory and Practice*, pages 129–159. MIT Press, 2000.

[21] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[22] Kalervo Järvelin and Jaana Kekäläinen. IR evaluation methods for retrieving highly relevant documents. In *ACM SIGIR Forum*, volume 51, pages 243–250, 2017.

[23] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. Recommendation independence. In *Conference on Fairness, Accountability and Transparency*, pages 187–201, 2018.

[24] C. Keribin, V. Brault, G. Celeux, and G. Govaert. Estimation and selection for the latent block model on categorical data. *Statistics and Computing*, 25(6):1201–1216, 2015.

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[26] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, Aug 2009.

[27] Aurore Lomet, Gérard Govaert, and Yves Grandvalet. Model Selection for Gaussian Latent Block Clustering with the Integrated Classification Likelihood. *Advances in Data Analysis and Classification*, 12(3):489–508, 2018.

[28] Benjamin M. Marlin, Richard S. Zemel, Sam T. Roweis, and Malcolm Slaney. Collaborative filtering and the missing at random assumption. In *Twenty-Third Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 267–275, 2007.

[29] Benjamin M. Marlin, Richard S. Zemel, Sam T. Roweis, and Malcolm Slaney. Collaborative filtering and the missing at random assumption. *CoRR*, abs/1206.5267, 2012.

[30] Rajesh Ranganath, Sean Gerrish, and David Blei. Black Box Variational Inference. In Samuel Kaski and Jukka Corander, editors, *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, pages 814–822, Reykjavik, Iceland, 22–25 Apr 2014. PMLR.

[31] Tim Räz. Group fairness: Independence revisited. *arXiv preprint arXiv:2101.02968*, 2021.

[32] Steffen Rendle, Li Zhang, and Yehuda Koren. On the difficulty of evaluating baselines: A study on recommender systems. *arXiv preprint arXiv:1905.01395*, 2019.

[33] Donald B. Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976.

[34] Sirui Yao and Bert Huang. Beyond parity: Fairness objectives for collaborative filtering. *CoRR*, abs/1705.08804, 2017.

[35] Ziwei Zhu, Xia Hu, and James Caverlee. Fairness-aware tensor-based recommendation. In *27th ACM International Conference on Information and Knowledge Management*, pages 1153—-1162, 2018.

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 81–90

DOI: 10.15439/2021F140
ISSN 2300-5963 ACSIS, Vol. 25

Rotation Variance in Graph Convolutional Networks

Nguyen Anh Mac
American School of Warsaw
Warszawska 202, 05-520 Bielawa, Poland
Email: 22mac_a@aswarsaw.org

Hung Son Nguyen
University of Warsaw
Krakowskie Przedmieście 26/28, 00-927 Warszawa, Poland
Email: son@mimuw.edu.pl

***Abstract*—Convolution filters in deep convolutional networks display rotation variant behavior. While learned invariant behavior can be partially achieved, this paper shows that current methods of utilizing rotation variant features can be improved by proposing a grid-based graph convolutional network. By performing spectral graph convolutions on features extracted from subareas of images, we are able to take advantage of the geometric nature of relational machine learning in graph neural networks to be able to overcome rotation variant features to perform object localization. We demonstrate that Grid-GCN heavily outperforms existing models on rotated images, and through a set of ablation studies, we show how the performance of Grid-GCN implies that there exist more performant methods to utilize fundamentally rotation variant features and we conclude that the inherit nature of spectral graph convolutions is able to learn invariant behavior.**

I. Introduction

OBJECT LOCALIZATION, i.e., specifying an object's location within an image, is an evolving subtask of computational vision problems that have grown in prevalence in recent years.

While the usage of deep convolutional networks in object localization shows near-human level accuracy [1], [2], convolutional neural networks exhibit fundamental flaws, primarily rotation variant behavior. This lack of rotation invariance is a cost of the translation invariance present within convolutional networks. Previous works in creating rotation invariant models utilize explicit methods in order to encourage or establish learned invariance within filters [3], [4]; however, it is clear that convolutional models do not naturally learn or exhibit rotation invariant performance.

Recent advancements in Graph Neural Networks allow us to utilize Graph Neural Networks (GNNs) in performing this object localization task with rotational invariant behavior. Relational machine learning presents an interesting methodology to approach these flaws of convolutional networks by leveraging the relational nature of the rest of the connected graph. In theory, this spectral characteristic of relational machine learning permits us to establish rotational invariance in an object localization system by utilizing message passing and neighborhood aggregation. There are previous works towards object localization with relational machine learning solutions [5], [6], [7], utilizing the geometric nature of graphs to represent images. In this paper, we present a system to perform grid-based object localization using Graph Neural Networks, titled Grid-GCN. Figure 1 demonstrates the usage and output of Grid-GCN. By employing spectral graph convolutions and pre-established feature extractors, we can effectively represent images geometrically and utilize neighborhood aggregation to learn rotational invariance. We show that our model performs comparably to state-of-the-art models (namely YOLOv4 and ResNest in grid localization) in general operation and outperforms these models on rotated images, despite lacking explicit methodology to counteract the impact of rotated images. Thus, we show that relational machine learning demonstrates the ability to learn invariant behaviour that deep convolutional networks are unable to, highlighting the current ineffective use of rotation variant features.

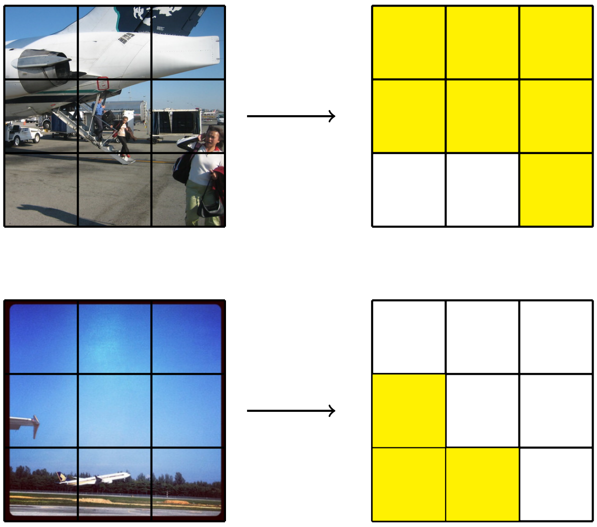


Fig. 1: **Output of Grid-GCN**. Grid-GCN outputs a grid containing the confidence that an object is present within the image. Yellow squares represent subareas where the confidence is above the detection threshold.

The structure of the manuscript is given as follows. In Section 2, we describe the previous works done regarding feature extraction, graph neural networks, and object localization. Section 3 outlines the principle concepts behind Grid-GCN and the theoretical and practical considerations of implementation.

In Section 4, we describe our experimental approach and the different datasets used to validate our approach, and Section 5 compares the results of Grid-GCN compared to existing object localization solutions. More importantly, in this section we

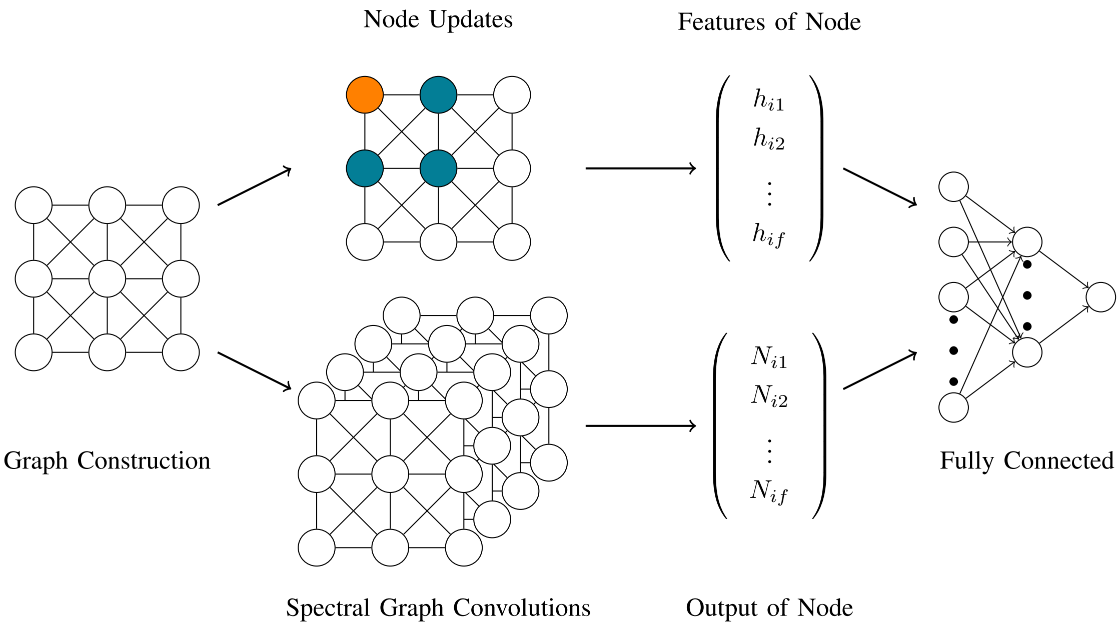


Fig. 2: **Grid-GCN process.** The figure depicts the process of Grid-GCN and Graph Neural Networks. Given a graph (constructed from an image, as described in 3.1 Graph Construction), inference begins with graph convolutions, as described in 3.2 Spectral Graph Convolutions for Grid-GCN. Graph convolutions produce an output vector for each node, inserted as the input for a series of fully connected layers (equivalent to classification layers in traditional neural networks). Standard Graph Neural Networks commence with node updating, where the orange node is updated with the features of the nodes connected (represented by blue nodes). For both processes, each node produces an output vector that represents the confidence that a node contains an object.

discuss the rotation variant nature of features from convolution filters and how Grid-GCN demonstrates that there exists better methods to use features to exhibit learned invariant behaviors. In Section 6, we conclude with further discussion of the practical and future works regarding this paper's findings.

II. Previous Work

A. Feature Extraction in Deep Learning

Feature extraction is the process of representing an initial set of information into a set of informative and non-redundant values called features. We primarily focus on feature extraction in deep learning models. Most, if not all, models relating to computer vision utilize a form of feature extraction [8], [1], [2], [9], [10], [11]. By learning on classifications of a specified object, the feature extraction model can effectively learn the necessary embeddings to represent specified objects' defining features. This paper employs the usage of a pre-trained ResNet model, which is detailed in [12]. ResNet has proven to be a reliable feature extractor for a multitude of domains [12], [13], and has shown the ability to operate at a range of resolutions [14], [15], [16], [17], which is crucial for the purposes of generalized datasets such as the Common Objects in Context dataset (abbreviated to COCO) [18].

B. Object Localization

Object localization refers to locating and indicating the location of objects within an image. There have been vast improvements in object localization in recent years due to significant technical improvements and the introduction of widely available and high-quality data sets such as the Common Objects in Context (COCO) dataset [18]. In this paper, we discuss object localization from the perspective of object detection and semantic segmentation. Historically, attempts at object localization in object detection tasks evaluate region proposals generated by heuristic algorithms [19]; however, region proposal networks and embedded-region detection methods have been steadily gaining prevalence [20], [10], [11]. Popular models, such as single-shot detectors [11], utilize an embedding region proposal network to perform localization, while models such as You-Only-Look-Once (YOLO) [21] utilize embedded region information generated from feature extraction to perform localization. Object localization, from the perspective of semantic segmentation, approaches localizing by indicating the presence of an object on a pixel-wise basis. Segmentation models [22], [23], [24] often build from convolutional networks to perform pixel-wise class predictions, essentially constructing a detailed localization of an image.

C. Graph Neural Networks (GNNs)

Relational machine learning, precisely graph neural networks, are becoming increasingly prevalent for solving computational vision tasks [5], [6], [7]. Solutions with relational machine learning often utilize the geometric nature of graphs to define a unique representation of images in order to effectively learn node relationships or graph classifications [25]. In this paper, we primarily focus on node classification, formally represented as learning the label of each node using the state of each node feature-vector to be as close as the ground truth of the node.

$$y_i = o(h_i, x_i) \tag{1}$$

where y_i is the output of a node, o is the output function, h_i is a node's state embedding, and x_i is the features of node i.

Modern GNNs follow a message propagation or neighbourhood aggregation principle, in which each embedding of each node updates depending on it's neighbors [25]. GNNs can be loosely categorized into a spectral filter and spatial filter models, both of which attempt to generalize convolution into a geometric manner, thus are dubbed 'Graph Convolutional Networks' (GCNs) [26]. However, while spatial approaches to relational machine learning exist in theory, it faces challenges regarding the representations of local neighborhoods [27], [28]. The key difference between spatial and spectral approaches in spatial approaches emphasizes edges in a node's nearest K-neighbours, while spectral approaches generalize across all neighbors effectively. The authors of [29] propose spectral graph convolutions in which filters are multiplied by graph signals and processed through graph coarsening to produce accurate representations of local neighborhoods. We employ the graph convolutional process given in [29] which is further explained in 3.2 Spectral Graph Convolutions, for the proposed solution.

III. Grid-GCN

Grid-GCN is inspired by the graph convolutional network model framework outlined in [26]. While we borrow ideas from [26], the organization and applications of the GCN method is completely novel. Traditional GNNs operate on the assumption that connected nodes are likely to share the same label. This assumption, however, while not necessarily incorrect, hinders modeling capacity. As such, GCNs removes this limitation by encoding a graph structure using a neural network and training on a supervised target. The readers are referred to [26] for more details.

The process outlined in this section consists of three stages: graph construction, feature extraction, and classification. In the case of this experiment, we view an image as a ten by ten grid. Grid-GCN uses a ten by ten grid due to the size constraints of the COCO dataset [18]. In general, it is vital to consider the amount of detail present within a subarea of a grid when deciding upon the resolution of the grid in order to convey succinct features relating to the object. The initial states of each subarea of this grid are set equal to the features of this subarea, and the input for each continuous iteration of internal inference is the initial states of each subarea.

Each node updates it's hidden state for a certain amount of steps (200 in our experiment). The output of each node are then placed into a traditional classification network, which returns a confidence measure. Figure 2 illustrates the outlined process.

A. Graph Construction

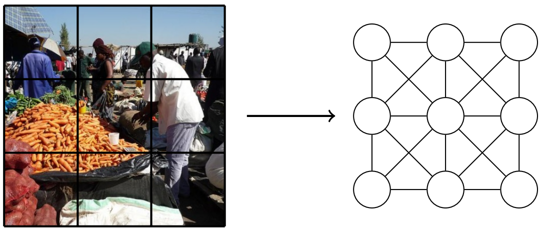


Fig. 3: **Graph Construction**. Images can be seen as a combination of subareas, which represent nodes of a graph. By doing so, we can represent an image geometrically effectively. Each node's initial state is the features of the subarea, and the weight of each connection is the cosine distance between respective initial states.

It is possible to view each image as a graph, where the nodes in the graph represent a subarea of a grid. The initial state of each node in this graph is set equal to the feature-vector extracted from the respective subarea. Figure 3 illustrates this concept. Each node is arbitrarily connected to its immediate vertical, horizontal, and diagonal neighbors on the grid structure whose edge weight is set equal to the cosine distance between the states of their respective nodes. By setting the edge weight equal to the cosine distance of the state of a node's immediate neighbor, the graph can comprehend the notion of scale and similarity. Nodes whose states are drastically different represent distinctions from background and foreground from one another; as such, their states should minimally impact each other. Formally, an edge between two nodes is determined as follows:

$$W_{ij} = \frac{s_i \cdot s_j}{||s_i|| ||s_j||} \tag{2}$$

where W_{ij} is the weight between nodes i and j, s_n is the state of the node n. It is important to note that since the graph is undirected, $W_{ij} = W_{ji}$.

By setting the edge of two nodes as the weight of the cosine distance of their states, objects are scaled-down, i.e., instances of an object whose size is equivalent to one subarea in the grid negatively impact neighboring nodes due to differing states. Likewise, nodes whose states are similar to one another reinforce the states of one another. By doing so, images whose objects span multiple subareas more directly influence one another.

B. Spectral Graph Convolutions

In this context, spectral convolutions offer a method to filter repeated information from a node's neighbors effectively. A node's state is not self-reinforced by an aggregate of the neighborhood with similar states but instead reinforced by a filtered state of the neighborhood, leading to diverse states during both message passing and classification. Diverse states preserve a node's original state while incorporating crucial information from the neighborhood. Mathematically, to perform effective convolutional operations on a graph, we must be able to effectively multiply a signal with a diagonalized filter on the Fourier basis. Convolution theory states that convolution in the spatial domain is equivalent to multiplication in the Fourier domain; thus, it is equivalent to eigendecomposition applied on the graph Laplacian.

$$F(x,\theta) = g_\theta(L)(x) = U g_\theta U^T x \tag{3}$$

where L is the normalized graph Laplacian $L = I_N - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$ (where I_N is the identity matrix, D is the diagonal degree matrix, and W is the weighted adjacency matrix), where $g_\theta(N) = diag(\theta)$ (where $\theta \; \varepsilon \; \mathbb{R}^N$ is a vector of Fourier coefficients), and U is the matrix of eigenvectors of the normalized graph Laplacian, and x is the state of the node [29], [26].

The naive approach to this process is generally considered too inefficient to be utilized in practical cases; as such, [29] proposed an efficient implementation of the above formula by parameterizing the filter with the normalized Laplacian graph and approximating the aforementioned function as a vector whose terms are a part of the Chebyshev polynomial.

$$g_\theta(\Lambda) = \sum_{k=0}^{K-1} \theta_k T_k(\overline{\Lambda}) \tag{4}$$

where $T_k(x)$ is the Chebyshev polynomial $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$ and $T_0 = 1$ and $T_1 = x$, and $\overline{\Lambda} = 2\Lambda{\lambda_{max}}^{-1} - I_n$ where λ_{max} represents the largest eigenvalue in L. This approximation reduces the computational complexity from $O(n^2)$ to $O(|\epsilon|)$ where ϵ is the set of edges. The reader is advised to [29] for a more detailed justification.

Consider the general equation for the features of a particular node at any given timestep t:

$$h_i^t = f(x_i^{t-1}, \bigcup_{\forall j:i\Rightarrow j} q(x_j^{t-1}, h_j^{t-1}, E_{ij})) \tag{5}$$

where f is the update function, q is the message preparation function, and E_{ij} are the features between nodes i and j. Graph Convolutional Networks specify the updating of the hidden states as follows:

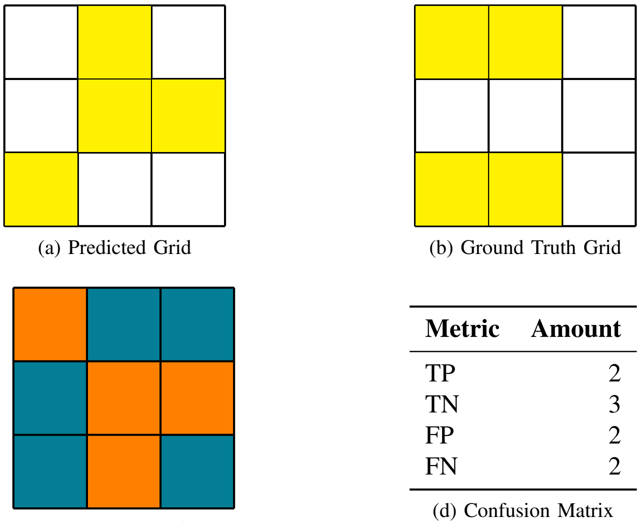
$$h_i^t = \sum_{\forall j:i\Rightarrow j} N_j^{t-1}\theta_j \tag{6}$$

where N_j is the output of a node from the convolution of the hidden state with a filter, approximated using the Chebyshev approximation, as defined as:

$$N_i^t = \sum_{k=0}^{K-1} \theta_k T_k(\overline{\Lambda}) \tag{7}$$

IV. The Proposed Method

It is important to note that grid localization, as outlined in the paper, is a novel task. Its nature is inherently difficult to correctly evaluate the success of grid-based localization due to a lack of previous work dedicated to the subject. As such, we define our own measures of success. We classify a subarea as predicted "correctly" (confidence score above the detection threshold) if an object or lack thereof matches the ground truth. Figure 4 illustrates an example of subareas of predicted grids are classified as a true positive, true negative, false positive, or false negative.



(a) Predicted Grid

(b) Ground Truth Grid

(c) Samples Grid

Metric	Amount
TP	2
TN	3
FP	2
FN	2

(d) Confusion Matrix

Fig. 4: **Comparing a predicted grid to the ground truth**. In the Predicted Grid and Ground Truth Grid, yellow boxes represent subareas with an (true or predicted) object. Thus, blue boxes within the Samples Grid represent true results (true positive and true negative), while orange boxes represents false results. True positive or true negative results occur when both the predicted and ground truth match and is positive or negative respectively. Similarly, false negative results occur when the predicted and ground truth subareas mismatch, and the ground truth subarea is positive. False positive results occur when the areas are mismatched, and the ground truth area is negative.

A. Measures of Evaluation

F-Score. F-Score is a classic metric used in many other works of object detection [30]. Classically, the F-Score measures the classification accuracy in terms of the harmonic mean of the precision and recall.

$$\begin{aligned} F\text{-}Score &= 2\frac{precision \cdot recall}{precision + recall} \\ &= \frac{2TP}{2TP + FP + FN} \end{aligned} \tag{8}$$

The harmonic mean property of the F-Score ensures that a balance of identifying positives and negatives is observed. This property is useful in scenarios with a large imbalance of positives and negatives. For example, consider the case where the rate of a positive sample to a negative sample is 1:99. A model that identifies all samples as negative returns an accuracy rate of 99%, however, is not particularly helpful in this context. Such a model, in contrast, has an F-Score of 0. In this experiment and practical applications of Grid-GCN, there is an imbalance of negative samples and positive samples (particularly skewed towards negative samples). Thus, F-Score is an appropriate measure of the success of Grid-GCN in relation to pre-existing models.

Matthews Correlation Coefficient. Matthews Correlation Coefficient (MCC) or phi coefficient, much like F-Score, is a popular statistic in machine learning [31], [32]. MCC is viewed as a balanced measure of true negative, true positive, false negative, and false positive samples. Specifically, MCC balances the performance of detecting each category of samples despite large imbalances in sample rates. MCC is defined as the following expression:

$$\frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{9}$$

The expression above produces results between -1 and 1, where -1 symbolizes the opposite correlation between prediction and truth, 0 is equivalent to random guessing, and 1 represents perfect predictions. While F-Score is an effective statistic that accurately depicts the performance of a model, there are particular imbalances in data that cause misleading results, primarily due to the fact that F-Score does not consider true negatives. For example, consider the scenario where a model detects 100 true positive results, five false positive results, one true negative, and nine false negative results. This would produce an F-Score of 93.46%, however, notice that the model is only detecting one negative sample out of a total of 6 (one true negative and five false positive samples). Thus, this model's MCC is 6.64%, reflecting the model's poor performance in detecting negative samples. Therefore, using MCC in conjunction with F-Score will accurately portray Grid-GCN's performance compared to other models.

B. Model Topology

The topology of Grid-GCN consists of three distinct, separable 'sections,' the feature extraction, graph convolution, and classification section. As mentioned previously, the feature extraction section was implemented with a pretrained Resnet-50, whose weights were frozen. The graph convolution section consists of two distinct spectral graph convolution layers. The function of these layers are detailed in 3.2 Spectral Graph Convolutions. We chose a filter size of three for this experiment, and hidden dimensions of 512 and 256 were chosen for the first and second convolution layers, respectively. A batch normalization and the RELU activation operator were applied to each of these layers. During the Grid-GCN training, a dropout of 0.5 was applied to the output of the second convolution layer. Dropout layers within GNNs have proven to be significantly important in the success of the generalization of models during training. The classification section of Grid-GCN consisted of two fully-connected layers of a hidden dimension of 256. Batch normalization and RELU were applied to the output of the first classification layer, while the activation function of the second layer was sigmoid. For each node, Grid-GCN outputs a vector representing the confidence of each class.

C. Model Configuration

For this paper, Grid-GCN is implemented using Pytorch [33] 1.7.0, Pytorch-Geometric [34] 1.6.3 and Tensorflow [35] 2.4.0 on 2 Nvidia K80 GPUs. For feature extraction, our experiment utilizes the Resnet-50 model trained on 'ImageNet' weights included in the Keras interface, and the fast spectral convolution operations described in 3.2 Spectral Graph Convolutions is implemented in Pytorch-Geometric by default. ResNet-50 uses a pooling mode of average, and we rescale the subareas in the grid to 224x224. The Adam [36] optimizer, included in Pytorch 1.7.0, whose hyperparameters were set to a learning rate of 0.01 and a weight decay of $5 \cdot 10^{-4}$ is used. Each image is processed and trained in 200 steps.

D. Dataset

We use the COCO Dataset [18] which is divided into three subsets, training, validation, and testing images, of which there are 118k images in training, 5k in validation, and 41k in testing. Each of these images is labeled using instances, of which each instance represents an object (with a predetermined class) and a mask. As such, we are able to translate the full masks in each image into a grid, where each subarea a scalar representing the presence of an object.

E. Training

Standard data augmentation procedures, consisting of random scaling, rotation, mirroring is applied. We trained the model using cross entropy loss, which has the following formula:

$$L(y, \hat{y}) = -\sum_{k}^{K} y^k \log \hat{y}^k \tag{10}$$

where K represents the number of nodes (of which there were 100).

Validation or testing labels is not used to train the model, and results mentioned in Table 1 are labels and samples exclusively from the COCO testing set.

Standard					Rotated (randomly)				
Model	Recall	Precision	F-Score	MCC	Model	Recall	Precision	F-Score	MCC
YOLOv4	0.7076	0.8398	0.7681	0.6001	YOLOv4	0.6851	0.8042	0.7399	0.5549
ResNest	0.6079	0.8486	0.7084	0.5406	ResNest	0.5171	0.7056	0.5969	0.3521
Grid-GCN	0.6941	0.8322	0.7569	0.5953	Grid-GCN	0.6871	0.8215	0.7483	0.5742

TABLE I: Recall, precision and F-Score of YOLOv4, ResNest, and Grid-GCN on the COCO dataset with and without rotation.

F. Baseline Models

YOLOV4. YOLOv4 [1] is a frontier object detection model, whose trained weights are made public[1]. Recently, YOLOv4 achieved an mean average precision (abbreviated to mAP) of 43.5% [1] on the COCO [18] dataset, which is the model that we utilized. While a newer version of the YOLO series (YOLOv5) has been made available, a technical report regarding the construction and improvements of YOLOv5 has not been made public, hence why YOLOv4 is used instead. Given that YOLOv4 is an object detection model which produces bounding boxes, several adjustments must be made. To compare these two models, we consider each subarea whose corners are within the predicted bounding box to be a positive result. Figure 5 illustrates comparing bounding boxes to a grid. Naturally, object detection models are disadvantaged in comparison to Grid-GCN in terms of Intersection-over-Union (IOU) simply due to the fact that object detection models are trained to perform localization in bounding boxes. The model incorporates additional subareas into the final positive sample count. Unfortunately, this is unavoidable as bounding boxes are the predominant solution to non-segmentation object localization. We create the final prediction of each subarea by assigning each unique class's values that intersects (or is present) within the subarea.

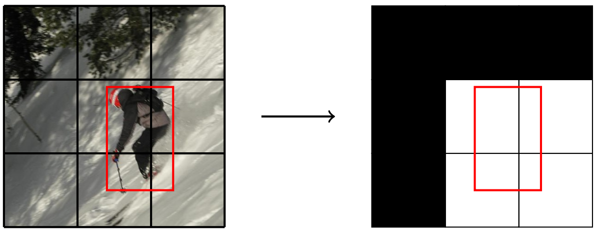


Fig. 5: **Comparing bounding boxes to a grid**, where white boxes represent positive subareas and black boxes represent negative subareas. The figure shows how object detection models are able to be compared to Grid-GCN by simply representing subareas as a collection of classes whose bounding box intersects the subarea.

ResNeSt. ResNeSt [23] is a state-of-the-art mode semantic segmentation model, achieving a mean Intersection over Union (mIoU) of 47.6% on the ADE20K validation set [23]. ResNeSt utilizes split attention networks and a unified computation block to outperform previous models in detection accuracy, classification accuracy, and computation time. At the time of writing, ResNeSt is the most accurate model (detection-wise) for the PASCAL-context dataset. For this experiment, we are utilizing ResNeSt-269, whose pretrained weights were made public [2]. Logistically, comparing the ResNeSt-269 model to Grid-GCN is trivial. We will perform a standard inference on the image (generating a mask) and divide it into an identical grid to the grid in Grid-GCN. Figure 6 illustrates a conversion from a mask to a grid.

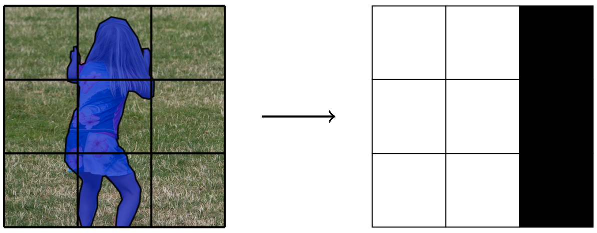


Fig. 6: **Masks on a grid**, where white boxes represent positive subareas and black boxes represent negative subareas. The figure shows how semantic segmentation models are able to be compared to Grid-GCN by simply representing subareas as positive if the mask overlaps. Image, with mask, is taken from COCO dataset's website using the explore feature[3].

V. Results

Standard operation. We define standard operation as inference and localization based on unmodified or unaugmentated images from the COCO dataset. As seen in Table 1, Grid-GCN performs comparably to YOLOv4 and significantly outperforms ResNest in all metrics. Low recall from the ResNest models suggests that the false negatives are significantly prevalent, though the poor behaviour of ResNest is difficult to explain.

Percent Decrease of Model Performance				
Model	Recall	Precision	F-Score	MCC
YOLOv4	3.18%	4.23%	3.67%	7.53%
ResNest	14.94%	16.85%	15.74%	34.87%
Grid-GCN	1.01%	1.29%	1.14%	3.54%

TABLE II: Percent decrease in performance of models where images were rotated randomly, on the COCO dataset.

Rotated Images. We perform a similar evaluation in this subcategory as the standard operation with the images rotated at random angles. YOLOv4, and approaches to computational

[1] https://github.com/pjreddie/darknet

[2] https://github.com/zhanghang1989/ResNeSt

[3] https://cocodataset.org/#explore

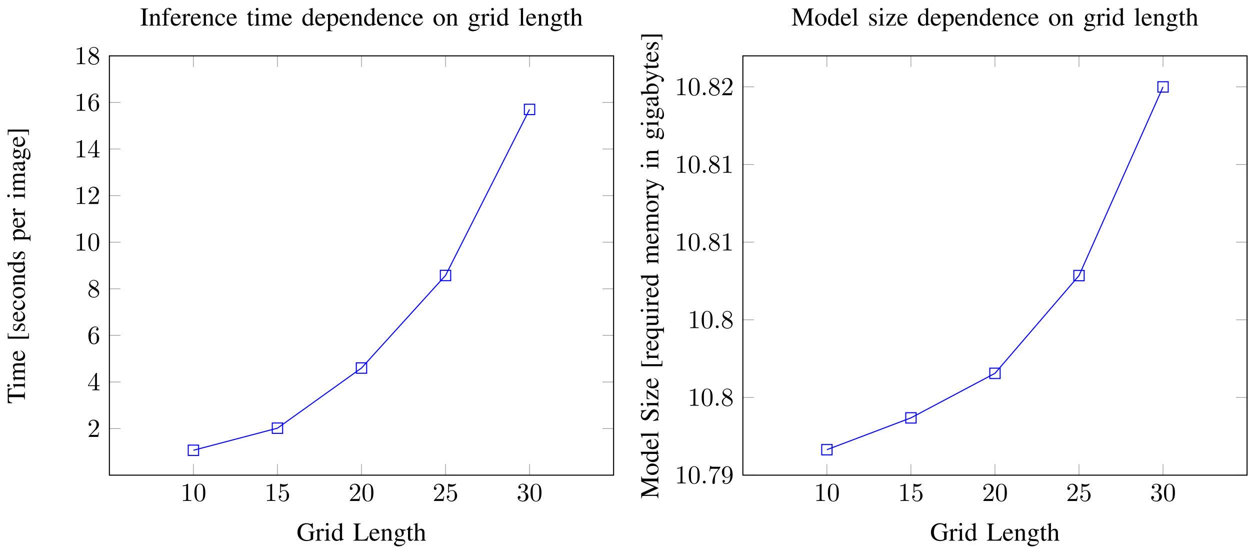


Fig. 7: **Model size and inference time dependence on grid length.** It is evident that both the size of the model (including graph information) and the inference time is exponentially correlated to the length of the grid. This exponential relationship is primarily driven by the fact that as grid size increases, the number of nodes in the graph increases exponentially.

vision tasks as a whole, rely on convolutional techniques to be able to reduce the dimensionality of data into a meaningful representation. It is known that convolutions, while being translation invariant do not exhibit rotational invariance. In more recent technology, the impact of rotational variance was reduced primarily by rotation augmentation in data, allowing filters to condition some form of rotational invariance. The impact of rotation is evident in both the YOLOv4 and ResNest models' performance after the images have been rotated, where there is a noticeable decrease in all performance metrics, as seen inTable 2. However, the diagonalization effect of bounding boxes may play a role in the decreased precision and recall for YOLOv4. Specifically, the diagonalization effect of bounding boxes increases the area of labeled positive samples, as the bounding box covers a more extensive than necessary area due to the nature of the output of such models. The diagonalization effect may explain the more pronounced decrease in precision for the YOLOv4 model. However, the drastic decrease in MCC suggests that the diagonalization effect was overshadowed by the increase in false negative results, implying that this diagonalization effect did not as heavily impact the performance of YOLOv4 as the rotational invariant nature of the model did.

While there was a decrease in recall and precision for Grid-GCN when images were rotated, this decrease was not as pronounced as the decrease for YOLOv4 and ResNest. Though this rotational invariance is not concrete (there was still a decrease in both recall and precision when images were rotated), the impact of rotation on Grid-GCN was significantly less than the impact on both YOLOv4 and ResNest, as shown in Table 2 by the lower percentage decrease in all performance metrics. It is important to emphasize that no explicit techniques to enforce rotational invariance were included within Grid-GCN. In other terms, Grid-GCN was able to learn a soft form of rotation invariance despite having rotation variant features. Previous works attempting rotation invariant image classification utilized explicit techniques [3], [4], whereas Grid-GCN learned rotation invariant behavior in an unsupervised fashion.

Grid-GCN's performance prompts the larger conversation of whether there is an aspect of features generated from convolution filters which contain a higher degree of rotation invariance than previously thought. Fundamentally, learned invariance in convolutional networks are caused by training pooling units which immediately proceed convolution filters [37]. These pooling layers allow a degree of rotation, in learned examples, to provide approximate or identical features from convolutional filters; hence demonstrating learned invariance. However, overabundance of pooling units causes loss of vital detail, thus, successful models which demonstrate learned invariance achieve a balance of pooling units to preserve both detail and invariance. According to their respective authors, all compared models and the ResNet backbone do achieve this balance [12], [1], [23]. Despite the input features of Grid-GCN being rotation variant, the comparatively minimal impact of rotation on Grid-GCN compared to both ResNeSt and YOLOv4 suggests there exists better methods to utilize learned variance in features.

To further understand the origin of Grid-GCN's learned invariance, we conducted a set of ablation studies to examine the behavior of our model. In the first ablation study, we removed the grid and feature extractor aspect of Grid-GCN, and we solely focused on if this invariant behavior originated from spectral graph convolutions. This was done by changing the size of the grid to 100 by 100, and resizing images to 100 by 100. Next, the initial states of nodes were not features extracted from ResNet, rather, normalized RGB values. After training this model under similar conditions, we compared the performance of such a model with standard and rotated images (labelled as "Non-Grid GCN" in the table below). In the second ablation study, we removed the spectral convolution aspect of the model. We still represented each image as a ten by ten grid with the initial state of each node still being output features from the ResNet backbone. Instead of using a GCN, these features were fed into a Support Vector Machine (SVM), and this SVM classified each node within the grid. After training this model under similar conditions, we compared the performance of such a model with standard and rotated images (labelled as "Grid SVM" in the table below)

Percent Decrease of Model Performance				
Model	Recall	Precision	F-Score	MCC
YOLOv4	3.18%	4.23%	3.67%	7.53%
ResNest	14.94%	16.85%	15.74%	34.87%
Grid-GCN	1.01%	1.29%	1.14%	3.54%
Non-Grid GCN	2.11%	3.02%	2.83%	5.74%
Grid SVM	5.33%	5.77%	4.14%	7.98%

TABLE III: Percent decrease in performance of models where images were rotated randomly, on the COCO dataset, including the modified GCN and Grid SVM.

In all metrics, Non-Grid GCN displayed less of a performance decrease in comparison to Grid SVM. This indicates that the impact of learned invariance from the ResNet backbone is not as impactful as the learned invariance originating from spectral graph convolutions. The exact reasons as to why this is the case is still unclear, however, a likely hypothesis is that it originates from the fact that adjacent nodes are taken into consideration during the spectral convolutions. Analogous to how convolutional networks learn invariant behavior due to pooling layers, it maybe be possible that the fact that the state of neighboring nodes are propagated act as a form of pooling. Furthermore, both of these models were more heavily impacted by rotation than Grid-GCN, suggesting that the minimal impact on the model's performance is attributed to both the rotation invariant behavior of the ResNet backbone and the dynamic nature of graph convolutions, not either/or. This implies that both the learned invariant behavior of the ResNet backbone and the consideration of neighboring nodes in spectral graph convolutions play a vital role in the learned invariant behavior of Grid-GCN.

In summary, Grid-GCN performs comparably to models in the metrics defined for this experiment (namely F-Score and MCC). Despite having no explicit rotation invariant aspects, Grid-GCN learned a soft-form of rotational invariant behavior and thus mitigated the impact of rotation variant filters on rotated images and outperformed state-of-the-art models on rotated images. It is likely that the origins of this learned invariant behavior results from a combination of the invariant behavior of the ResNet backbone and the dynamic nature of graph convolutions.

VI. Discussion

The most major limitation of Grid-GCN is the nature of grid localization. For the sake of demonstrating the fact that rotation variant features could be better utilized, Grid-GCN performs object localization on a grid (justification for such a choice is further explained in the previous section). Practically speaking, this design choice heavily limits the use cases for Grid-GCN. While there exists tasks which require grid-based object localization, namely subtasks of object detection [38], [39], [40], grid-based object localization by itself is rare. However, it is important to consider the fact that the primary goal of this paper was to emphasize the idea that current models are ineffective at utilizing rotation variant features and that graph neural networks are able to display invariant behavior. Grids are limited yet necessary drawback to use graphs in computer vision.

Another consideration for practical uses of Grid-GCN is the inference time per image. On average, Grid-GCN spends 1068 ms per image, while YOLOv4 has an inference time of 155 ms, and ResNest has an inference time of 407 ms with the implementation outlined in Section 4.3. This vast imbalance of time between compared models and Grid-GCN can be attributed to the graph construction. In the process of graph construction, feature extraction of a subarea occurs n^2, 100 in the case of this experiment, times. Moreover, the process of graph convolutions is, in its nature, slower than existing filters and standard convolutions as each image require a series of steps on processing (200 in the case of this experiment), thus, why Grid-GCN is significantly slower than the compared models. Ideally, developing a method to parallelize feature extraction across subareas (as they are independent of one another) may offer a significant improvement to inference time. Future works for accelerating graph processing on the hardware level will also offer a significant improvement of inference time [41].

Another factor is the scalability of the model. As the resolution of the grid increases, the resources (namely memory and time) required to sustain the model increase exponentially. Figure 7 illustrates both the inference time per image and the memory required to sustain the model as the grid length increases. Though the model itself does not grow exponentially (the model's memory processing is static), the graph information grows exponentially. Thus, the exponential nature of grids prohibits high-resolution versions of Grid-GCN in a practical manner both in time constraints and in-memory constraints.

VII. Conclusion

We have introduced a grid-based relational learning framework for object localization using graph convolutional networks. We show that our framework was able to display rotational invariant behavior, outperforming state-of-the-art object localization models on rotated planes despite lacking explicit methodology to enable invariant performance. Thus, we show that GCNs are able to implicitly learn invariant behaviour that deep convolutional networks are unable to. Moreover, we discuss the origins of the learned invariant behavior in Grid-GCN, namely considering spectral graph convolutions and the ResNet backbone through a set of ablation studies.

Our paper highlights two distinct topics of interest for future work. The methodology in which relational machine learning learns invariant behaviour in a more effective manner than traditional machine learning frameworks is still unclear. Exploring this methodology can be done by modifying and alternating any aspect of the Grid-GCN process. Moreover, viability of the outlined processes is dependent on alleviating time and memory constraints associated with graph machine learning.

References

[1] A. Bochkovskiy, C. Wang, and H. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *CoRR*, vol. abs/2004.10934, 2020. [Online]. Available: https://arxiv.org/abs/2004.10934

[2] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 261–318, Oct. 2019. [Online]. Available: https://doi.org/10.1007/s11263-019-01247-4

[3] J. Kim, W. Jung, H. Kim, and J. Lee, "Cycnn: A rotation invariant cnn using polar mapping and cylindrical convolution layers," 2020.

[4] D. Marcos, M. Volpi, and D. Tuia, "Learning rotation invariant convolutional filters for texture classification," *2016 23rd International Conference on Pattern Recognition (ICPR)*, Dec 2016. [Online]. Available: http://dx.doi.org/10.1109/ICPR.2016.7899932

[5] W. Shi and R. Rajkumar, "Point-GNN: Graph neural network for 3d object detection in a point cloud," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jun. 2020. [Online]. Available: https://doi.org/10.1109/cvpr42600.2020.00178

[6] A. Luo, X. Li, F. Yang, Z. Jiao, H. Cheng, and S. Lyu, "Cascade graph neural networks for RGB-D salient object detection," in *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XII*, ser. Lecture Notes in Computer Science, A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, Eds., vol. 12357. Springer, 2020, pp. 346–364. [Online]. Available: https://doi.org/10.1007/978-3-030-58610-2_21

[7] Y. Wang, K. Kitani, and X. Weng, "Joint object detection and multi-object tracking with graph neural networks," in *Proceedings of (ICRA) International Conference on Robotics and Automation*, May 2021.

[8] A. O. Salau and S. Jain, "Feature extraction: A survey of the types, techniques, applications," in *2019 International Conference on Signal Processing and Communication (ICSC)*. IEEE, Mar. 2019. [Online]. Available: https://doi.org/10.1109/icsc45622.2019.8938371

[9] S. Khalid, T. Khalil, and S. Nasreen, "A survey of feature selection and feature extraction techniques in machine learning," in *2014 Science and Information Conference*. IEEE, Aug. 2014. [Online]. Available: https://doi.org/10.1109/sai.2014.6918213

[10] Z. Chen, X. Jin, B. Zhao, X. Wei, and Y. Guo, "Hierarchical context embedding for region-based object detection," in *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXI*, ser. Lecture Notes in Computer Science, A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, Eds., vol. 12366. Springer, 2020, pp. 633–648. [Online]. Available: https://doi.org/10.1007/978-3-030-58589-1_38

[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," *Lecture Notes in Computer Science*, p. 21–37, 2016. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-46448-0_2

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jun. 2016. [Online]. Available: https://doi.org/10.1109/cvpr.2016.90

[13] D. Rukhovich, K. Sofiiuk, D. Galeev, O. Barinova, and A. Konushin, "Iterdet: Iterative scheme for object detection in crowded environments," in *Structural, Syntactic, and Statistical Pattern Recognition - Joint IAPR International Workshops, S+SSPR 2020, Padua, Italy, January 21-22, 2021, Proceedings*, ser. Lecture Notes in Computer Science, A. Torsello, L. Rossi, M. Pelillo, B. Biggio, and A. Robles-Kelly, Eds., vol. 12644. Springer, 2020, pp. 344–354. [Online]. Available: https://doi.org/10.1007/978-3-030-73973-7_33

[14] H. Touvron, A. Vedaldi, M. Douze, and H. Jégou, "Fixing the train-test resolution discrepancy," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, Eds., 2019, pp. 8250–8260. [Online]. Available: https://proceedings.neurips.cc/paper/2019/hash/d03a857a23b5285736c4d55e0bb067c8-Abstract.html

[15] Z. Lu, X. Jiang, and A. Kot, "Deep coupled ResNet for low-resolution face recognition," *IEEE Signal Processing Letters*, vol. 25, no. 4, pp. 526–530, Apr. 2018. [Online]. Available: https://doi.org/10.1109/lsp.2018.2810121

[16] M. Kawulok, P. Benecki, S. Piechaczek, K. Hrynczenko, D. Kostrzewa, and J. Nalepa, "Deep learning for multiple-image super-resolution," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 6, p. 1062–1066, Jun 2020. [Online]. Available: http://dx.doi.org/10.1109/LGRS.2019.2940483

[17] A. Zhou, Y. Ma, Y. Li, X. Zhang, and P. Luo, "Towards improving generalization of deep networks via consistent normalization," *CoRR*, vol. abs/1909.00182, 2019. [Online]. Available: http://arxiv.org/abs/1909.00182

[18] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," in *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, ser. Lecture Notes in Computer Science, D. J. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., vol. 8693. Springer, 2014, pp. 740–755. [Online]. Available: https://doi.org/10.1007/978-3-319-10602-1_48

[19] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*. IEEE Computer Society, 2014, pp. 580–587. [Online]. Available: https://doi.org/10.1109/CVPR.2014.81

[20] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017. [Online]. Available: https://doi.org/10.1109/TPAMI.2016.2577031

[21] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 779–788. [Online]. Available: https://doi.org/10.1109/CVPR.2016.91

[22] A. Tao, K. Sapra, and B. Catanzaro, "Hierarchical multi-scale attention for semantic segmentation," *CoRR*, vol. abs/2005.10821, 2020. [Online]. Available: https://arxiv.org/abs/2005.10821

[23] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, Z. Zhang, H. Lin, Y. Sun, T. He, J. Mueller, R. Manmatha, M. Li, and A. J. Smola, "Resnest: Split-attention networks," *CoRR*, vol. abs/2004.08955, 2020. [Online]. Available: https://arxiv.org/abs/2004.08955

[24] Y. Yuan, X. Chen, and J. Wang, "Object-contextual representations for semantic segmentation," in *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part VI*, ser. Lecture Notes in Computer Science, A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, Eds., vol. 12351. Springer, 2020, pp. 173–190. [Online]. Available: https://doi.org/10.1007/978-3-030-58539-6_11

[25] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020. [Online]. Available: https://doi.org/10.1016/j.aiopen.2021.01.001

[26] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. [Online]. Available: https://openreview.net/forum?id=SJU4ayYgl

[27] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2014. [Online]. Available: http://arxiv.org/abs/1312.6203

[28] Q. Liu, M. Kampffmeyer, R. Jenssen, and A. Salberg, "SCG-Net: Self-Constructing Graph Neural Networks for Semantic Segmentation," *CoRR*, vol. abs/2009.01599, 2020. [Online]. Available: https://arxiv.org/abs/2009.01599

[29] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, Eds., 2016, pp. 3837–3845. [Online]. Available: https://proceedings.neurips.cc/paper/2016/hash/04df4d434d481c5bb723be1b6df1ee65-Abstract.html

[30] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, H. Omata, T. Kashiyama, and Y. Sekimoto, "Global road damage detection: State-of-the-art solutions," in *IEEE International Conference on Big Data, Big Data 2020, Atlanta, GA, USA, December 10-13, 2020*, X. Wu, C. Jermaine, L. Xiong, X. Hu, O. Kotevska, S. Lu, W. Xu, S. Aluru, C. Zhai, E. Al-Masri, Z. Chen, and J. Saltz, Eds. IEEE, 2020, pp. 5533–5539. [Online]. Available: https://doi.org/10.1109/BigData50022.2020.9377790

[31] D. Chicco and G. Jurman, "The advantages of the matthews correlation coefficient (MCC) over f1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, Jan. 2020. [Online]. Available: https://doi.org/10.1186/s12864-019-6413-7

[32] S. Boughorbel, F. Jarray, and M. El-Anbari, "Optimal classifier for imbalanced data using matthews correlation coefficient metric," *PLOS ONE*, vol. 12, no. 6, p. e0177678, Jun. 2017. [Online]. Available: https://doi.org/10.1371/journal.pone.0177678

[33] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[34] M. Fey and J. E. Lenssen, "Fast graph representation learning with pytorch geometric," *CoRR*, vol. abs/1903.02428, 2019. [Online]. Available: http://arxiv.org/abs/1903.02428

[35] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: A system for large-scale machine learning," in *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, ser. OSDI'16. USA: USENIX Association, 2016, p. 265–283.

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[37] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[38] J. Gao, T. Zhang, and C. Xu, "Graph convolutional tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[39] A. Nicolicioiu, I. Duta, and M. Leordeanu, "Recurrent space-time graph neural networks," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/383beaea4aa57dd8202dbff464fee3af-Paper.pdf

[40] Y. Chen, Y. Kalantidis, J. Li, S. Yan, and J. Feng, "A^2-nets: Double attention networks," *CoRR*, vol. abs/1810.11579, 2018. [Online]. Available: http://arxiv.org/abs/1810.11579

[41] A. Auten, M. Tomei, and R. Kumar, "Hardware acceleration of graph neural networks," in *2020 57th ACM/IEEE Design Automation Conference (DAC)*, 2020, pp. 1–6.

DOI: 10.15439/2021F128
ISSN 2300-5963 ACSIS, Vol. 25

Query Specific Focused Summarization of Biomedical Journal Articles

Akshara Rai, Suyash Sangwan, Tushar Goel, Ishan Verma, Lipika Dey
TCS Research
New Delhi, India
Email: (akshara.rai, suyash.sangwan, t.goel, ishan.verma, lipika.dey)@tcs.com

***Abstract*—During COVID-19, a large repository of relevant literature, termed as "CORD-19", was released by Allen Institute of AI. The repository being very large, and growing exponentially, concerned users are struggling to retrieve only required information from the documents. In this paper, we present a framework for generating focused summaries of journal articles. The summary is generated using a novel optimization mechanism to ensure that it definitely contains all essential scientific content. The parameters for summarization are drawn from the variables that are used for reporting scientific studies. We have evaluated our results on the CORD-19 dataset. The approach however is generic.**

***Index Terms*—Extractive Summarization, Query Answering, Biomedical Text-mining, Scientific Repositories, CORD-19 dataset**

I. Introduction

With the rapid rise of scholarly articles in the biomedical domain, there has been a growing urgency to explore Natural Language Processing (NLP) techniques that can process vast volumes of content to generate intelligent insights, which can then be selectively explored by the experts. This was proved once again during the current COVID-19 pandemic. There has been a stupendous rise in related biomedical articles that have been published over the period. While it undoubtedly helped medical practitioners, virologists, immunologists, policy makers, public health planners, drug manufacturers and many others associated to healthcare services, it also highlighted the need for efficient mechanisms to enable intelligent navigation through this sea of content. The needs of end-users can be quite varied in nature. For example, in the current scenario, while medical professionals need insights about drugs and procedures, a virologist would be interested in studying the nature of the virus and hence look for literature reporting the virus's transmission, incubation, susceptibility to external factors, etc. Public policy makers, on the other hand, need information to design effective policies and guidelines to keep the spread controlled. Since, time is premium for every user, a mechanism that will enable the user to grasp the key aspects covering objectives, methodology and findings or outcomes, if any, of an article is an important ask from the NLP community.

In May 2020, as Allen Institute shared a large repository called "CORD-19"[1], which contained bio-medecial articles related to corona virus. Kaggle further announced a challenge, in which some of the key questions asked by the end-users were put up for the natural language processing community to find out efficient methods to answer them. A two-way communication ensued on the platform between the end-users and the NLP researchers, wherein the focus was to understand the requirements clearly. The discussion led to clearer elicitation of information components from different categories of users. As it turned out, while the information components were different for different category of users, all users wanted to view the relevant findings about the components in a contextual way, that would make it easy for them to interpret the significance of the results. For example, epidemiologists specifically wanted to know the "incubation period" of the virus, in order to design policies for prevention and control. However, as different values for incubation period were reported by scientists from different corners of the world, the epidemiologist wanted the result to be presented along with its context that included the sample type, sample size and most importantly the statistical outcomes of these results. The contextual presentation was clearly needed to help them decide whether to accept or reject the results. Similarly, a doctor may want to know about the drugs that were found to be effective, but along with it also the details about patient condition and treatment course, to help in decision making. It has to be further remembered that a single article may contain information that could be of interest to multiple categories of users, though all of it may not be of interest to any one category. Though the requirements were first published in Kaggle, subsequently, TREC also posted similar requirements from the CORD-19 collection. For a large number of short queries, it posted additional narratives stating stricter requirements for a retrieved article to qualify as relevant. It was observed that the narratives were similar to the user requirements mentioned in the Kaggle platform.

Motivated by the above requirements, in this paper, we present a mechanism that can create a query-specific contextually focused summary of an article for the end-user. The rationale of the proposed mechanism comes from commonly followed reporting style for bio-medical articles, especially for reporting experimental studies and case studies. The target of our work is to generate a uniformly-structured summary that contains all relevant information for a specific end-user. Thus, two end-users, based on their requirements, may see two

[1] https://allenai.org/data/cord-19

different summaries of the same document, though both the summaries will be structured in a similar fashion. Section II presents more details about the structure of an ideal summary.

This is achieved in three stages.

- We first provide a query representation mechanism that can accommodate the user requirements in terms of 5 parameters that comprise key aspects of a scientific study: *study type, sample size, sample type, measures/results, evidence of measure*. The rationale for selecting these five parameters is explained in detail in section III.
- Next, an optimization-driven method is proposed to select a minimal set of sentences that can satisfy the requirements of a query. It is done by scoring the sentences based on their information content with respect to the above-mentioned parameters, with additional constraints imposed on their proximity. The proximity constraints have been designed based on commonly followed practices for reporting outcomes in bio-medical scientific publications. These sentences form a "snippet", which can provide the key outcomes at a glance. This is explained in section IV-C.
- Finally, a contextual summary creation method is proposed. The contextual summary is created by rearranging the set of sentences selected by the optimizer and augmenting them with additional content, if necessary, to create a cohesive and comprehensive summary. This is explained in section IV-D.

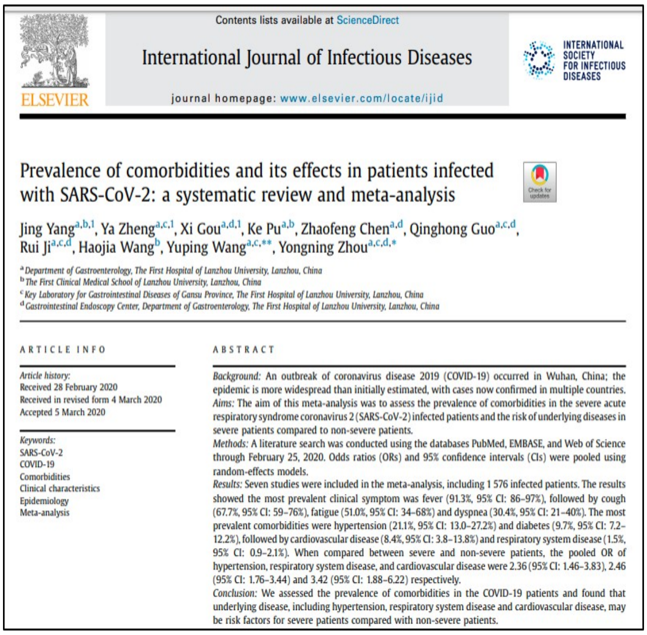
  The proposed approach ensures that the necessary information components found in the documents are always contained in the summary.

In the absence of any gold-standard data-set for evaluating the contextually focused summaries created by the proposed method, we have evaluated the summaries by comparing them with the abstracts provided along with the articles. We show that, for journals that insist on a structured summary for authors, the generated summaries are very similar to author-provided summaries. However, such journals are very few. Thus only 25% articles in the repository were found to have high-quality author-generated structured summaries. The focused summary generation method can thus be used to generate high quality summaries for a larger collection of bio-medical articles. This, by itself, is a very significant contribution to the domain of bio-medical literature analysis. The results and observations are discussed in detail in section V.

It may be noted that, the proposed mechanism is not an alternative to online document search systems which pull documents from an indexed collection in response to a query. Rather, our work is intended to augment the search results by generating a query specific summary for all articles retrieved by the search engine in response to a query. Subsequently, documents are re-ranked based on the quality of the summary. The contextual summary can be shown as a snippet to the end-user for faster comprehension.

A summary of related work in the allied area has been presented in section VI.

II. Structure of an Ideal Summary of a Bio-medical article



Contents lists available at ScienceDirect

International Journal of Infectious Diseases

journal homepage: www.elsevier.com/locate/ijid

Prevalence of comorbidities and its effects in patients infected with SARS-CoV-2: a systematic review and meta-analysis

Jing Yang[a,b,1], Ya Zheng[a,c,1], Xi Gou[a,d,1], Ke Pu[a,b], Zhaofeng Chen[a,d], Qinghong Guo[a,c,d], Rui Ji[a,c,d], Haojia Wang[b], Yuping Wang[a,c,**], Yongning Zhou[a,c,d,*]

[a] *Department of Gastroenterology, The First Hospital of Lanzhou University, Lanzhou, China*
[b] *The First Clinical Medical School of Lanzhou University, Lanzhou, China*
[c] *Key Laboratory for Gastrointestinal Diseases of Gansu Province, The First Hospital of Lanzhou University, Lanzhou, China*
[d] *Gastrointestinal Endoscopy Center, Department of Gastroenterology, The First Hospital of Lanzhou University, Lanzhou, China*

ARTICLE INFO

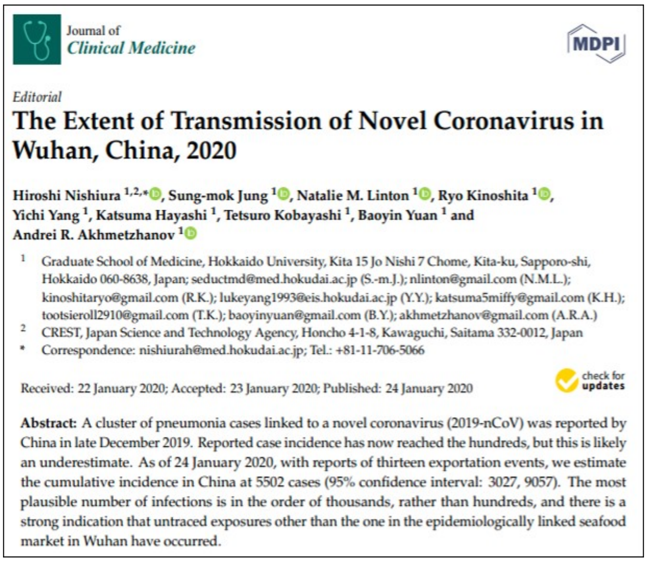
Article history:
Received 28 February 2020
Received in revised form 4 March 2020
Accepted 5 March 2020

Keywords:
SARS-CoV-2
COVID-19
Comorbidities
Clinical characteristics
Epidemiology
Meta-analysis

ABSTRACT

Background: An outbreak of coronavirus disease 2019 (COVID-19) occurred in Wuhan, China; the epidemic is more widespread than initially estimated, with cases now confirmed in multiple countries.
Aims: The aim of this meta-analysis was to assess the prevalence of comorbidities in the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) infected patients and the risk of underlying diseases in severe patients compared to non-severe patients.
Methods: A literature search was conducted using the databases PubMed, EMBASE, and Web of Science through February 25, 2020. Odds ratios (ORs) and 95% confidence intervals (CIs) were pooled using random-effects models.
Results: Seven studies were included in the meta-analysis, including 1 576 infected patients. The results showed the most prevalent clinical symptom was fever (91.3%, 95% CI: 86–97%), followed by cough (67.7%, 95% CI: 59–76%), fatigue (51.0%, 95% CI: 34–68%) and dyspnea (30.4%, 95% CI: 21–40%). The most prevalent comorbidities were hypertension (21.1%, 95% CI: 13.0–27.2%) and diabetes (9.7%, 95% CI: 7.2–12.2%), followed by cardiovascular disease (8.4%, 95% CI: 3.8–13.8%) and respiratory system disease (1.5%, 95% CI: 0.9–2.1%). When compared between severe and non-severe patients, the pooled OR of hypertension, respiratory system disease, and cardiovascular disease were 2.36 (95% CI: 1.46–3.83), 2.46 (95% CI: 1.76–3.44) and 3.42 (95% CI: 1.88–6.22) respectively.
Conclusion: We assessed the prevalence of comorbidities in the COVID-19 patients and found that underlying disease, including hypertension, respiratory system disease and cardiovascular disease, may be risk factors for severe patients compared with non-severe patients.

Fig. 1: Structured abstract



Journal of *Clinical Medicine*

Editorial

The Extent of Transmission of Novel Coronavirus in Wuhan, China, 2020

Hiroshi Nishiura [1,2,*], Sung-mok Jung [1], Natalie M. Linton [1], Ryo Kinoshita [1], Yichi Yang [1], Katsuma Hayashi [1], Tetsuro Kobayashi [1], Baoyin Yuan [1] and Andrei R. Akhmetzhanov [1]

[1] Graduate School of Medicine, Hokkaido University, Kita 15 Jo Nishi 7 Chome, Kita-ku, Sapporo-shi, Hokkaido 060-8638, Japan; seductmd@med.hokudai.ac.jp (S.-m.J.); nlinton@gmail.com (N.M.L.); kinoshitaryo@gmail.com (R.K.); lukeyang1993@eis.hokudai.ac.jp (Y.Y.); katsuma5miffy@gmail.com (K.H.); tootsieroll2910@gmail.com (T.K.); baoyinyuan@gmail.com (B.Y.); akhmetzhanov@gmail.com (A.R.A.)
[2] CREST, Japan Science and Technology Agency, Honcho 4-1-8, Kawaguchi, Saitama 332-0012, Japan
[*] Correspondence: nishiurah@med.hokudai.ac.jp; Tel.: +81-11-706-5066

Received: 22 January 2020; Accepted: 23 January 2020; Published: 24 January 2020

Abstract: A cluster of pneumonia cases linked to a novel coronavirus (2019-nCoV) was reported by China in late December 2019. Reported case incidence has now reached the hundreds, but this is likely an underestimate. As of 24 January 2020, with reports of thirteen exportation events, we estimate the cumulative incidence in China at 5502 cases (95% confidence interval: 3027, 9057). The most plausible number of infections is in the order of thousands, rather than hundreds, and there is a strong indication that untraced exposures other than the one in the epidemiologically linked seafood market in Wuhan have occurred.

Fig. 2: Unstructured abstract

A well-structured summary is expected to contain all required information in a compact, cohesive and comprehensible fashion. Though scientific documents usually contain abstracts that present a short and concise summary of the document, our analysis of the CORD-19 collection revealed that abstracts vary widely in size and nature, depending on the journal in which it is published. We observed that bio-medical documents contain two types of abstracts, i.e. 'Structured abstract' and 'Unstructured abstract'. Structured abstracts usually present a well-defined and detailed summary of the document. Figure 1 shows an example of a structured abstract [1], where *Background, Method, Results and Conclusion* of the experiment are separately presented in the abstract itself. Unstructured abstracts, as shown in Figure 2 [2], on the other hand are generally short and may not convey all the important elements included in the introduction, method, or findings sections. Both these abstracts were created by the respective authors, who selected which information goes to the abstract and which does not. In the absence of a strict requirement, the author-created abstract may or may not contain the information that is required by a user, even though it may be contained in the article.

The proposed work intends to cover this gap by providing a mechanism to create focused well-structured summaries on the fly, which will contain the user-required information, if it is there in the document. These summaries should be similar in form to the structured abstracts shown in Figure 1. In order to do that, we exploit the inherent structure that is observed in the published articles. Bio-medical articles usually follow a specific format for reporting their findings. The findings are usually reported along with additional details about (a). the type of the study or the way the experiment or study was conducted (b). details about the subject of the experiment i.e. about the sample types, categorization of the samples, sample size etc. (c). results of experiments or observations (d). evidence of measure for different sample categories (e). the significance of the results. There is also a discipline that is maintained while reporting these items. For example, significance of a result is explained along with evidence of measure.

In the next section, we first present a few sample queries published on the Kaggle site along with the requirements of each. Subsequently, we discuss how these requirements can be mapped to the scientific parameters and converted to a slot-value format, which is used later to construct optimization constraints. The optimizer then uses these constraints to select an optimal set of sentences that can satisfy the user requirements.

III. Query representation mechanism for summary construction

Table I shows four types of questions, posted under different task categories in the CORD-19 [2] challenge by various groups of users. Each question is accompanied by a narrative that specifies what kind of information is required from the documents, to answer the queries. These four questions represent four broad and exhaustive categories, which cover most of the user queries posed to the collection. We now present a mapping of these queries to the parameter requirements mentioned earlier. The mapping is done to five different slots that can be associated to specific types of values.

1) *Study Type:* describes a broad category for the type of work reported in the document. It could be a systematic review, a case study or case series, a simulation study or an experimentation. This covers almost all kind of documents, but more may be added.
2) *Sample Size:* is used to define the size of the study population, samples studied or papers reviewed to compute the result. For example, 50 Patients, 120 case reports, etc.
3) *Sample Type:* describes the sub sample of the population addressed or the type of samples that were considered for the study. For example, population addressed can be pregnant women, children, elderly, smokers, etc.
4) *Measures/Results:* These are the quantitative outcomes or findings presented in a document after analysis of the data. They can be statistical findings like odds ratio, hazard risk, etc. on potential risks or other outcomes like drug effectiveness, prevalence, etc.
5) *Evidence of Measure:* These are additional qualifiers or filters that are applied on the measures/results to quantify the level of evidence. Evidences can be expressed in terms of sets of sub-samples generated from the population. For example, the risk posed by COVID-19 to smokers can vary depending on their age and other co-morbidities present. The impact of a policy or guideline depends on the country it is implemented at. Thus, these elements can be used to present the evidence of measure of various queries.

Table II presents a few sample user queries from Kaggle site, along with their mapping to the question type presented in Table I, further slotted according to the type of information required. The slot-value requirements for each question type is derived from the narratives. This is further validated using the target requirements mentioned for these queries at the Kaggle site.

Slot items are associated with factor-specific constraints that are designed to ensure that only meaningful information components are picked up. For example, odds ratio is usually specified in a paper as "OR <INTEGER>, 95% CI <RANGE>", incubation period is presented as "number of days", country names can only be from a set of known entities, drug names can be recognized using Biological entity taggers. Each slot is also associated to an encapsulated information extraction procedure which hunts for feasible values for that slot. Table II also gives some examples of accepted study design types for the bio-medical domain. A list of such constraints has been curated from available literature and data on the challenge sites. This list can be extended.

In order to ensure the coverage of queries using these categories of questions and slot types, we have additionally considered the queries presented by the TREC challenge makers[3] to be addressed from the CORD-19 collection. We were able to map approximately 67% queries to these 4 broad categories mentioned in Table I and further identify the slot requirements on the basis of the narrative. For example,

[2] https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge/tasks

[3] we have used the topic set from Round 1 data - https://ir.nist.gov/covidSubmit/data.html

Type	**Category**	**Kaggle Questions**	**Detailed Requirement**
1	Risk Factors	What do we know about COVID-19 risk factors?	Data on potential risks factors: Smoking, pre-existing pulmonary disease, Co-infections and other co-morbidities. Severity of disease, including risk of fatality among symptomatic hospitalized patients, and high-risk patient groups. Susceptibility of this specific population. Mitigation measures that could be effective for control
2	Epidemiological Requirements/Clinical characteristics	What is known about transmission, incubation, and environmental stability?	What do we know about natural history, transmission, and diagnostics for the virus? What have we learned about infection prevention and control? Range of incubation periods for the disease in humans (and how this varies across age and health status) and how long individuals are contagious, even after recovery. Prevalence of asymptomatic shedding and transmission. Persistence and stability on a multitude of substrates and sources (e.g., nasal discharge, sputum, urine, fecal matter, blood). Persistence of virus on surfaces of different materials (e,g., copper, stainless steel, plastic).
3	Treatment/Diagnostics Efficacy	What do we know about vaccines and therapeutics?	Effectiveness of drugs being developed and tried to treat COVID-19 patients. Clinical and bench trials to investigate less common viral inhibitors against COVID-19. Capabilities to discover a therapeutic (not vaccine) for the disease, and clinical effectiveness studies to discover therapeutics, to include antiviral agents. Use of diagnostics such as host response markers (e.g., cytokines) to detect early disease or predict severe disease progression, which would be important to understanding best clinical practice and efficacy of therapeutic interventions.
4	Non Pharmaceutical Intervention/ Relevant External Factors	What do we know about non-pharmaceutical interventions and the Relevant factors related to COVID-19	Rapid design and execution of experiments to examine and compare NPIs currently being implemented. Rapid assessment of the likely efficacy of school closures, travel bans, bans on mass gatherings of various sizes, and other social distancing approaches. Methods to control the spread in communities, barriers to compliance and how these vary among different populations. Models of potential interventions to predict costs and benefits that take account of such factors as race, income, disability, etc. Seasonality of transmission, How does temperature and humidity affect the transmission of 2019-nCoV? Significant changes in transmissibility in changing seasons? Effectiveness of personal protective equipment (PPE)

TABLE I: User given questions and their detailed requirements.

	Inferred Slot Requiremnts					
Sample User Query	Query Type	Study Type	Sample Size	Sample Type	Measures/Results	Evidence of Measure
Risk to pregnant women	1	Systematic review, case series	#patients	Pregnant women	Odds ratio, hazard ratio, severity	-
Incubation period of Sars-Cov-2	2	Simulation, meta-analysis	#patients	-	No. of Days/weeks	age, gender
Effectiveness of Remdesivir in treating COVID-19	3	RCT, systematic review, meta-analysis	#patients	patients treated with Remdesivir	Odds ratio, hazard ratio, severity	Therapeutic method(s) utilized/assessed
Effect of social distancing in reducing virus spread	4	Simulation, Cross-sectional study, systematic review	-	Population(general, healthcare, minority)	Percentage Decrease/Increase, mortality rate, days	Intervention: Social Distancing, Geographical location, model used

TABLE II: Slotted user requirements for sample Kaggle queries

the TREC query- *'Are patients taking Angiotensin-converting enzyme inhibitors (ACE) at increased risk for COVID-19?'* can be mapped to question Type 1 with the following constraint - <Sample Type, Patients taking ACE inhibitors>. Similarly, another query *'How long can the coronavirus live outside the body'* can be assigned to Type 2 with <Measure/Results, Persistence(in days, hrs, half life)> and <Evidence of Measure, (Sample Observed, Detection Method)> slot requirements. Table III shows a few examples of queries from the TREC dataset.

The remaining 33% queries required theoretical evidence based excerpts, such as - *'What are best practices in hospitals and at home in maintaining quarantine?', 'How has lack of testing availability led to under reporting of true incidence of Covid-19?'*.

A. Ensuring consistency of information

After identifying the required slots, they are further bucketed together to ensure meaningful information extraction. The buckets represent groups of slot items that are inter-dependent on each other with respect to the given query. The inter-dependence of these items is either expressed as a linguistic constraint or a proximity constraint. These constraints are also parsed from the narrative. For example, for a query "what is the range of incubation period for different age groups?" the slot value pairs are filled up as <Measure/Results, incubation period>, <Evidence of measure, age group> and <Sample size, #patients>. This in turn implies that the statement "The average incubation period was 4 days," found in a document wouldn't be complete. It needs additional information for the result to be accepted. A sentence found in close proximity to the above one was "We considered 157 confirmed cases, aged 44-60 years, 74 female (47.1%) and 38 imported cases (24.2%)." A complete snippet would have to contain both the sentences. By adding Measures/Results along with the Evidence of Measure in a single bucket, we can generate a more comprehensive and coherent snippet for the user query. Additionally, information about whether it was a simulation experiment or a systematic review, i.e. the study type of the document is also presented in the snippet. This is independent of the final result being reported and is therefore added in a separate bucket. Thus, there can be two buckets in the ar-

	Inferred Slot Requiremnts					
Sample TREC Question & Narrative	Query Type	Study Type	Sample Size	Sample Type	Measures/Results	Evidence of Measure
Are patients taking Angiotensin-converting enzyme inhibitors (ACE) at increased risk: interactions between coronavirus and angiotensin converting enzyme 2 (ACE2) receptors, risk for patients and recommendations for these patients.	1	Systematic review, case series, Retrospective	#patients	patients taking ACE	Odds ratio, hazard ratio, severity	-
How long does coronavirus remain stable on surfaces: SARS-CoV-2's virus's survival in different environments (surfaces, liquids, etc.) outside the human body while still being viable for transmission to another human	2	Experimental study, Systematic review,meta-analysis	#samples	-	No. of Days/weeks, half life	Surfaces, # studies, Method used
What types of rapid testing for Covid-19 have been developed? : ways to diagnose Covid-19 more rapidly	3	clinical trial, retrospective, systematic review, meta-analysis	#patients	infected patients	Efficiency, speed of Assay	Detection Method: Rapid
How does the coronavirus respond to changes in the weather: virus viability in different weather/climate conditions, transmission of the virus in different climate conditions	4	Simulation, Cross-sectional study, systematic review, retrospective	-	-	Incidence, transmission, mortality rates	External Factor: weather, Geographical location, model used

TABLE III: Sample TREC questions and narratives with slotted user requirements

rangement to capture all the slots. Bucket 1 contains (Evidence of Measure, Measures/Result) and Bucket 2 includes (Sample type, Sample Size, Study Type).

The buckets can be interpreted as context provider for the information components to ensure that randomly occurring strings or values of a certain type are not accepted just because of a keyword match.

IV. Query Focused Summary Generation

The task of of generating query specific focused summaries is carried out in phased manner. Initially, the slot-value pairs are searched within the document collection. This is done by locating the entities within the document. Each document is subjected to a pre-processing phase for the purpose. After that, a set of minimal number of sentences is selected that satisfies the slot requirements, with additional constraints imposed on the proximity of slots within a single bucket, using integer linear programming. A subsequent phase of enhanced document summarization is carried out to present the information in a coherent and comprehensible form.

A. Document Pre-processing

Like all document processing tasks, search for information is preceded by a one-time activity that comprises of document pre-processing and information extraction. Each document in the set is passed through a pre-processing pipeline for cleaning and tokenizing it into sentences using SciSpaCy [3]. Each sentence is then indexed according to its unique document-id and the section label where it belongs in the document. Each document is then subjected to the following processes-

- **Biomedical Entity Extraction**: Given the biomedical documents, this module extracts biomedical entities like Participant Age, Participant Sex, Participant Sample size, Participant Condition, Surgical Intervention, Physical Intervention, Educational Intervention, Psychological Intervention, Control Intervention, Outcome Physical, Pain Mentions, Mortality Mentions, Mental States and Adverse effects. These entities are extracted using a BERT-based sequence labelling approach described in [4]. Additionally, biomedical entities like DNA, Cell Type, Protein, Chemical, Organ names, Drug, etc. are also extracted using SciSpaCy.
- **Named Entity Extraction**: Named entities like name of the locations, person, organizations, expressions of quantities ('0.2 ng/mL'), time ('less than 24 hours'), age ('49 years old', 'one week old') are extracted from each document using SpaCy [5].
- **Sentence embedding generation**: Sentence embeddings are also generated using Facebook's Infersent pre-trained encoder [6] to create a 300-dimensional vector for a sentence. It uses Bidirectional LSTM with max pooling to capture the context and generic information available for a variety of tasks. These embeddings capture the semantics of a sentence better by embedding the context in the encoding.

B. Mechanism for sentence scoring

In this section, we present how the specific information components required for a query are located within the documents and scored to generate a snippet. First, the sentences are checked for the presence of any of the required slot values. Slot specific search methods are deployed for this purpose. The extraction methods commonly used for the different slots are as follows: -

1) *Measures/Results-* As observed from the summary tables provided by the CORD-19 challenge makers, values fitting this slot (like OR, p-value, HR, etc) follow a

set pattern, which can be expressed using a regular expression such as "<MeasureName>= <INTEGER>(, 95% CI <RANGE>)?". At first, we used a regular expression matching algorithm to extract instances of this type. But the pattern matching approach resulted in noisy extractions and also missed certain instances that varied slightly from this pattern. Therefore, we moved on to use a BiLSTM-CRF sequence tagger [7] to identify the measures/results in sentences, which showed an accuracy of 97%. Here, we have used the results from above pattern-matching approach along with certain hand tagged instances (that were not detected earlier) to create the annotated training data for a sequence tagger. We have excluded the noisy extractions of pattern-matching approach from the training data. Since the task is to identify a set of literals/token following a pattern, we did not use any sequence tagging algorithm requiring semantic context.

2) *Study Type-* These are pre-specified strings and keywords found in text. A comprehensive design dictionary curated by a team of epidemiologists has been provided to help the CORD-19 research community for effective retrieval. [4]
3) *Sample Size-* This is extracted by tagging *'Participant Sample Size'* instances in text using the biomedical entity extractor described in the previous section.
4) *Sample Type-* Values are extracted using the biomedical entity extraction module. For any given query, findings like patient condition, patients undergoing any surgical intervention, patients having any drug administered, etc. can be selected for this slot depending on the requirement. For example, for the query 'risk to cancer patients due to COVID-19' - <patient condition, 'Cancer'> is added to the slot. For 'effectiveness of hydroxychloroquine in treatment of COVID-19 patients' the slot-value pair <Drug, 'Hydroxychloroquine'> is added.
5) *Evidence of Measure-* Values are extracted using the biomedical and Named entity extraction modules explained in the previous section. Extractions like Patient Age, Gender, country, etc. are included in this slot.

Any sentence that contains at least one value is retained for scoring, while the remaining ones are assigned a score of 0. The final score assigned to a sentence depends on three factors, which are explained below-

Confidence score from sentence type - The section headers of the document are also taken into account while scoring sentences. Thus, sentences from "review" section score less than those coming from other sections of the document, since the latter are considered to be fundamental contributions from the document under consideration. Since, section headers are not always unambiguous, special checks are put into place to check for reference and citation patterns as well as linguistic constructs to identify such sentences. For computing the confidence value, sentences from "review" sections are penalized by a value of (ρ), and the findings fundamental to the document are rewarded with (ρ), such that $0<\rho<1$.

Intra-bucket score - Sentences containing values for certain slots also gain for being in proximity of other sentences containing values in the same bucket. As a corollary, between two sentences that contain values for the same slot, the one that contains additional values for other slots belonging to the same bucket will score higher. This is referred to as intra-bucket score of a sentence.

Inter-bucket score – Sentences also gain some reward from being in proximity to other sentences that contain values for slots from other buckets. The inter bucket proximity ensures that the overall context of all the findings remains consistent.

We now present the scoring equations.

Proximity between two sentences S_i and S_j, is computed as an inverse function of the distance between the sentences in the document and also takes into account their corresponding section headers.

$$Proximity(S_i, S_j) = \frac{(1+section\_reward(i,j))}{(1+distance(S_i,S_j))} \tag{1}$$

where, distance (S_i, S_j) = abs (position (S_i) - position(S_j)), position(S_i) indicates original sentence number of S_i, and Section_reward (i,j) = 1, if the section header of sentences is same; otherwise 0.

Let V = $\{v_1, v_2, v_3, \ldots, v_m\}$ be the set of values required by the query. Then the scores for a sentence S_i having a value v_k is expressed as follows:

$$Intra\_Bucket\_Score = \sum_k (Confidence(v_k) + \\ \sum_p (max(Proximity(S_i, S_j)))) \tag{2}$$

$\forall$ v_k, v_p ϵ V, s.t. bucket(v_p) = bucket(v_k),

$\forall$ j s.t. S_j is the closest sentence that contains a value for a slot v_p that belongs to the same bucket, including itself.

$$Inter\_Bucket\_Score = \sum_k (Confidence(v_k) + \\ \sum_p (max(Proximity(S_i, S_j)))) \tag{3}$$

$\forall$ v_k, v_p ϵ V, s.t. bucket(v_p) $\neq$ bucket(v_k) $\forall$ j s.t. S_j is the closest sentence that contains a value for a slot v_p that belongs to a different bucket, including itself.

Score (S_i) is now computed as-

$$Score(S_i) = \alpha(Intra\_Bucket\_Score(S_i)) + \\ (1-\alpha)(Inter\_Bucket\_Score(S_i)), \tag{4}$$

We take α >0.5 to give more weightage to the Intra_Bucket scores over the Inter_Bucket scores. The sentence score is then normalized s.t. Score (S_i) ϵ [0,1].

[4] https://docs.google.com/spreadsheets/d/1t2e3CHGxHJBiFgHeW0dfwtvCG4x0CDCzcTFX7yz9Z2E/edit#gid=1217643351

C. Optimal snippet generation

Our goal is now to use the above scores to identify the minimal set of sentences that can form a snippet.

Let us suppose that query Q has 'm' slot values divided into different buckets. Let S = $\{S_1, S_2, \ldots, S_n\}$ be the set of sentences which have a non-zero scores after scoring. The following optimization algorithm finds the minimal set of sentences that contain all the 'm' values, if present.

Let VS(i, j) = 1, if value v_j is found in S_i; otherwise 0.

Let x(i) = 1, if S_i is selected in optimal snippet and 0 otherwise

Then the objective function for the optimization problem is expressed as-

Objective Function:

$$Maximize \sum_i (x(i) * (Score(S_i) - 1)) \tag{5}$$

Subject to constraints:

$$\sum_i (VS(i,j) * x(i)) >= 1 \;\; \forall v_j \, found \, in \, D \tag{6}$$

$$\sum_i x(i) <= |V| \tag{7}$$

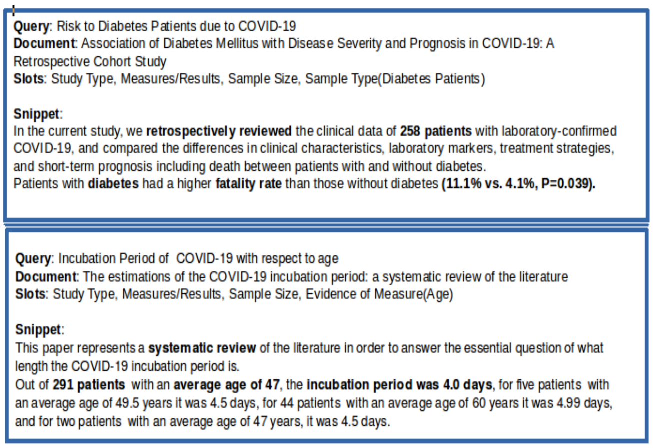
$$\sum_i x(i) >= 1 \tag{8}$$

The value (–1) is added to ensure that minimum number of sentences are finally selected. The constraint in equation 6 ensures that at least 1 sentence is picked to cover each value, provided that value is reported by the document D. Finally, equations 7 and 8 enforce that at least 1 sentence is selected from the document and maximum number of sentences selected are no more than the type of values required to address the user given query. This is solved using Integer Linear programming.

Figure 3 shows the snippet generated using the above optimization approach for two documents [8, 9], along with the slot values for the queries 'Risk to Diabetes Patient' and 'Incubation period with respect to age'. It can be seen from these examples that the individual sentences by themselves are not enough. Reporting *'Fatality rate was 11.1%'* doesn't convey the confidence of the finding. By additionally reporting <*Patient Condition, 'Diabetes'*>, <*Sample Size, '258 Patients'*> *and* <*Study Type', 'Retrospective'*>, a much better picture can be presented. The second example also highlights how the proximity constraint helps provide maximum information in minimum sentences, making it much more comprehensible.

D. Contextual focused summary generation

In this section, we present an enhanced summarization approach which generates a fixed length extractive summaries for documents, by checking for sentence representativeness along with the scores from the previous section. For each candidate sentence to be included in the summary, it's 300 – dimensional vector embedding is created using Infersent.



Query: Risk to Diabetes Patients due to COVID-19
Document: Association of Diabetes Mellitus with Disease Severity and Prognosis in COVID-19: A Retrospective Cohort Study
Slots: Study Type, Measures/Results, Sample Size, Sample Type(Diabetes Patients)

Snippet:
In the current study, we **retrospectively reviewed** the clinical data of **258 patients** with laboratory-confirmed COVID-19, and compared the differences in clinical characteristics, laboratory markers, treatment strategies, and short-term prognosis including death between patients with and without diabetes.
Patients with **diabetes** had a higher **fatality rate** than those without diabetes **(11.1% vs. 4.1%, P=0.039).**

Query: Incubation Period of COVID-19 with respect to age
Document: The estimations of the COVID-19 incubation period: a systematic review of the literature
Slots: Study Type, Measures/Results, Sample Size, Evidence of Measure(Age)

Snippet:
This paper represents a **systematic review** of the literature in order to answer the essential question of what length the COVID-19 incubation period is.
Out of **291 patients** with an **average age of 47**, the **incubation period was 4.0 days**, for five patients with an average age of 49.5 years it was 4.5 days, for 44 patients with an average age of 60 years it was 4.99 days, and for two patients with an average age of 47 years, it was 4.5 days.

Fig. 3: Snippets generated for queries along with slot values

Sentence score (Sc) for the ith sentence in the jth document is generated as follows -

$$\begin{aligned} Sc(S_i^j) = {} & Sc_{Rank}(S_i^j) + Sc_{Title}(S_i^j) + \\ & Sc_{Position}(S_i^j) + Sc_{Domain}(S_i^j), \end{aligned} \tag{9}$$

where Sc_{Rank} (S_i^j) is the representativeness score assigned using the TextRank algorithm, by checking the sentence's similarity with all other sentences, using the corresponding Infersent vectors. Sc_{Title} (S_i^j) is computed using the cosine similarity between the title and sentence vectors. Position score proves to be very effective in document summarization as it is a good indicator of significant sentences and is computed as

$$Sc_{Position}(S_i^j) = \frac{Len_j}{Pos_i * (Len_j - Pos_i + 1)}, \tag{10}$$

where, Len_j is the length of jth document, and Pos_i is the position of ith sentence in the document.

Sc_{Domain} (S_i^j) denotes the score computed in the earlier section based on the slot requirements. All these scores are normalized and added to give us the final sentence score.

In order to remove redundancy, we use an algorithm similar to the MMR algorithm [10], that focuses on ensuring diversity in the sentences being selected. The sentences are sorted based on the decreasing value of their scores Sc (S_i^j) and the highest scored sentence is selected to be included in the final summary first. The next sentences are selected based on the following conditions:

Sentences are added to the final summary, iff the cosine similarity of the sentence with the selected set of sentences is below a threshold β.
Sentences having similarity with a selected sentence greater than the threshold β are discarded if they belong to the same section in the document.

This process is repeated for all the remaining sentences, till selected sentence count reaches a maximum count τ.

To ensure that the summaries are connected and coherent, the selected sentences are re-ordered according to their position in the document. Preserving document order guarantees that the summary has sentences from the aim and introduction presented first, followed by the methodology and finally, the results and conclusions.

V. Experiments and Evaluations

A. Dataset description

The Covid-19 Open Research Dataset (CORD-19) is a collection of scientific papers on Covid-19, SARS-CoV-2, and related historical coronaviruses. The dataset contains a primary metadata file containing unique paper id, author, journal, publication date, abstract etc. and link to full-text file name. Full texts are available for some files in json format.

B. Snippet evaluation and observations

We have conducted the evaluation of the snippet generation system on recently-published articles from the CORD-19 dataset. Due to lack of gold standard data, the evaluation was done manually for 10 queries across 4 categories (Table I), on 500 documents. We consider Study type, Sample Size, Evidence of Measure, Sample Type and Measures/Results as the required slots and compare the findings with the values reported in the abstracts, also measuring the overall correctness with respect to the document as well. The manual inspection of generated snippets with respect to the documents showed that study type and sample size were retrieved correctly 70.2% and 67.4% times respectively. Out of these, it was observed that 82.24% and 66.52% of the times these values matched with study type and sample size reported in the abstract. For measures/results (i.e. the quantitative findings), we evaluate them as correct if the extraction is reported in association with the user query/keyword. We observed that of the 73.6% correctly extracted values only 26.3% of the snippet values matched with the findings in the abstract. In 47.5% cases we observed that the abstracts either did not report any statistical findings or reported findings were not relevant to the query. This could be because the main theme of the document was different from that of the query. This further emphasizes the need for generating snippets and summaries from documents that answer the user queries.

C. Evaluating summaries- results and observations

We use Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [11] scores for evaluating the summaries. It determines the quality of a summary automatically, by comparing it to human (ideal) generated summaries (we use the abstracts as model summaries here). ROUGE-N (unigram and bigram match) and ROUGE-L (Longest Common Subsequence match) scores were chosen for our experiments.

The generated summaries were grouped based on the type of abstract (structured and unstructured) in the document. We observed that only 145 documents (25.3%), out of 573 scientific documents summarized, had structured abstracts, remaining documents either had no abstract or had an unstructured one.

We have generated two different types of summaries using TextRank algorithm, as shown in the Table IV. In the baseline approach, we have generated generic summaries, using $Sc_{Rank}(S_i^j)$, Sc_{Title} (S_i^j), and $Sc_{Position}$ (S_i^j) scores. But in our final approach (i.e. Contextual focused summary), we have incorporated user requirements by using Sc_{Domain} (S_i^j) for scoring sentences.

In order to determine the performance, results are also compared with some existing text summarization algorithms, like LSA [12] and TextRank [13]. It can be seen from Table IV that our system performs better than these summarization algorithms. There is a 6.9% increase in ROUGE-L scores after including Sc_{Domain} (S_i^j) score in case of structured abstracts. High ROUGE scores with structured abstracts indicates that the summaries generated by our method have been able to cover the important information and findings well. Unstructured abstracts, on the other hand, seldom include results or description of the methodology. By including slots like Study Design, Sample Size, Statistical Measure/Results, the summaries generated by our approach become more informative and can present facts and details that are mostly not covered in the abstracts.

Figure 4 shows the relation between number of words in abstract and the ROUGE scores for documents with unstructured abstracts. Since the longer abstracts are supposed to be more detailed and informative, it can be seen that with the increase in word count, the ROUGE scores also increase. The evaluation with low word count abstract provides a reverse indicator for measuring the quality of the summaries, as a lower overlap means that a lot of additional information has been captured in the summary as well, that was missing in the abstract.

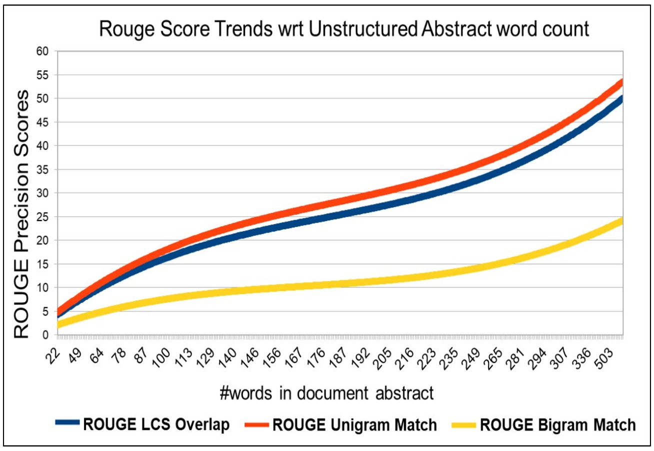


Fig. 4: ROUGE scores trends with respect to the Abstract word count

VI. Related Work

Text summarization has attracted the attention of NLP researchers for a long time. Latent Semantic Analysis (LSA) based approach was introduced in [12], which uses a singular value decomposition on word-sentence matrix. This

Approach	Documents with Structured Abstracts			Documents with Unstructured Abstracts		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
TextRank	0.41009	0.16537	0.37369	0.29361	0.11551	0.26283
LSA	0.40491	0.11812	0.36868	0.34119	0.09503	0.30515
Baseline	0.45208	0.17312	0.41852	0.39170	0.16082	0.35838
Baseline+Domain Scores	0.47615	0.19624	0.44744	0.38696	0.15809	0.35438

TABLE IV: F-measure for summaries generated (length = 10 sentences)

way sentences that discuss important topics are chosen as candidates for summaries. One of the most successful text summarization systems called TextRank [13] was introduced in 2004. TextRank uses a graph-based algorithm similar to PageRank [14], in which similarity between two sentences is computed in terms of their content overlap. Later, [15] enhanced TextRank and proposed the use of longest common substrings based cosine distance between pairs of sentences. BM25 [16] can also be used as a ranking function to retrieve the candidate sentences for the summary. Single-document summarization approach was proposed in [17], that maximizes concept coverage using Integer Linear Programming(ILP). They also presented a weighing method for combining position to emphasize important concepts.

The information available for clinicians and clinical researchers is growing exponentially, both in the biomedical literature and patients' health records. We need strategies to cope with this information overload as biomedical literature provides clinicians and clinical researchers with a valuable source of knowledge to assess the latest advances, develop and validate new hypotheses, conduct experiments, and interpret their results [18, 19].

Several approaches have been proposed for summarization in biomedical domain. The applications mainly include summarizing treatments [20], summarizing drug information [21], summarizing clinical reports [22], and electronic health records [23]. One such work is presented in [24], a graph-based summarizer that uses the Unified Medical Language System (UMLS) to identify concepts and the semantic relations between them to construct a semantic graph that represents the document. A degree-based clustering algorithm was then used to identify different themes or topics within the text. Authors in [25] proposed a clustering and itemset mining based Biomedical Summarizer (CIBS) that also utilize UMLs to map text to concepts and then passes it to an itemset mining algorithm, for topic extraction. Sentences are clustered and related sentences from within these clusters are selected to produce a summary.

Text summarization approaches focusing on answering user queries are particularly of interest as it can aid medical practitioners identify salient and relevant information. The work in [26] presented one such approach that utilizes labeled data that is publicly available, pre-trained medical domain word embeddings along with a set of simple features for generating query focused extractive summaries.

Query-based text summarization based on common-sense knowledge and word sense disambiguation was proposed in [27]. Their technique finds semantic relatedness score between query and input text document for extracting relevant sentences. It finds correct sense of each word of a sentence with respect to the context of the sentence and hence provides query-relevant summaries.

VII. Conclusion

In this paper, we present summarization mechanism that can create a query-specific contextually focused summary of an article for the end-user. Initially, a query representation mechanism is defined that can accommodate the user requirements in terms of a fixed number of parameters that comprise key aspects of a scientific study. Further, an optimization-driven mechanism is used for retrieving minimal number of sentences relevant to an elaborate scientific query. These sentences form a snippet which provides the key outcomes at a glance. Finally, a contextual summary is created by rearranging the set of sentences selected by the optimizer and augmenting them with additional content. The target of the current work is to generate a uniformly-structured summary that contains all relevant information for a specific end-user. Thus the summaries are customized to the needs of the user. The results have been evaluated using ROUGE scores. The summaries generated by the proposed method have high ROUGE scores with the author-written summary, whenever one is present. For the remaining documents, the generated summary is a useful addition. From an application point of view, we believe that our snippet generation and summarization approach can be easily applied to other data sets by updating the slot requirements.

In future, we would like to explore more on the document structures, sentence type classification and abstractive summarization approaches for reducing the information overload even further. We also intend to extend the methods to work for any scientific document collection, beyond bio-medical literature. We are also evaluating it for a larger set of queries with enough variation in their structures and design automated evaluation mechanisms, since getting manual feedback is difficult.

References

[1] J. Yang, Y. Zheng, X. Gou, K. Pu, Z. Chen, Q. Guo, R. Ji, H. Wang, Y. Wang, and Y. Zhou, "Prevalence of comorbidities and its effects in patients infected with sars-cov-2: a systematic review and meta-analysis," *International Journal of Infectious Diseases*, vol. 94, pp. 91–95, 2020.

[2] H. Nishiura, S.-m. Jung, N. M. Linton, R. Kinoshita, Y. Yang, K. Hayashi, T. Kobayashi, B. Yuan, and A. R.

Akhmetzhanov, “The extent of transmission of novel coronavirus in wuhan, china, 2020,” 2020.

[3] M. Neumann, D. King, I. Beltagy, and W. Ammar, “Scispacy: Fast and robust models for biomedical natural language processing,” *arXiv preprint arXiv:1902.07669*, 2019.

[4] T. Dasgupta, I. Mondal, A. Naskar, and L. Dey, “Extracting semantic aspects for structured representation of clinical trial eligibility criteria,” in *Proceedings of the 3rd Clinical Natural Language Processing Workshop*. Online: Association for Computational Linguistics, Nov. 2020. doi: 10.18653/v1/2020.clinicalnlp-1.27 pp. 243–248. [Online]. Available: https://www.aclweb.org/anthology/2020.clinicalnlp-1.27

[5] M. Honnibal and I. Montani, “spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing,” 2017, to appear.

[6] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, “Supervised learning of universal sentence representations from natural language inference data,” *arXiv preprint arXiv:1705.02364*, 2017.

[7] Z. Huang, W. Xu, and K. Yu, “Bidirectional lstm-crf models for sequence tagging,” *arXiv preprint arXiv:1508.01991*, 2015.

[8] Y. Zhang, Y. Cui, M. Shen, J. Zhang, B. Liu, M. Dai, L. Chen, D. Han, Y. Fan, Y. Zeng *et al.*, “Association of diabetes mellitus with disease severity and prognosis in covid-19: a retrospective cohort study,” *Diabetes research and clinical practice*, vol. 165, p. 108227, 2020.

[9] N. Zaki and E. A. Mohamed, “The estimations of the covid-19 incubation period: a systematic review of the literature,” *medRxiv*, 2020.

[10] J. Carbonell and J. Goldstein, “The use of mmr, diversity-based reranking for reordering documents and producing summaries,” in *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 1998, pp. 335–336.

[11] C.-Y. Lin, “Rouge: A package for automatic evaluation of summaries,” in *Text summarization branches out*, 2004, pp. 74–81.

[12] J. Steinberger and K. Jezek, “Using latent semantic analysis in text summarization and summary evaluation,” *Proc. ISIM*, vol. 4, pp. 93–100, 2004.

[13] R. Mihalcea and P. Tarau, “Textrank: Bringing order into text,” in *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004, pp. 404–411.

[14] L. Page, S. Brin, R. Motwani, and T. Winograd, “The pagerank citation ranking: Bringing order to the web.” Stanford InfoLab, Tech. Rep., 1999.

[15] F. Barrios, F. López, L. Argerich, and R. Wachenchauzer, “Variations of the similarity function of textrank for automated summarization,” *arXiv preprint arXiv:1602.03606*, 2016.

[16] S. Robertson and H. Zaragoza, *The probabilistic relevance framework: BM25 and beyond.* Now Publishers Inc, 2009.

[17] H. Oliveira, R. Lima, R. D. Lins, F. Freitas, M. Riss, and S. J. Simske, “A concept-based integer linear programming approach for single-document summarization,” in *2016 5th Brazilian Conference on Intelligent Systems (BRACIS)*. IEEE, 2016, pp. 403–408.

[18] R. Smith, “Strategies for coping with information overload,” 2010.

[19] F. Davidoff and J. Miglus, “Delivering clinical evidence where it’s needed: building an information system worthy of the profession,” *Jama*, vol. 305, no. 18, pp. 1906–1907, 2011.

[20] H. Zhang, M. Fiszman, D. Shin, C. M. Miller, G. Rosemblat, and T. C. Rindflesch, “Degree centrality for semantic abstraction summarization of therapeutic studies,” *Journal of biomedical informatics*, vol. 44, no. 5, pp. 830–838, 2011.

[21] M. Fiszman, T. C. Rindflesch, and H. Kilicoglu, “Summarizing drug information in medline citations,” in *AMIA Annual Symposium Proceedings*, vol. 2006. American Medical Informatics Association, 2006, p. 254.

[22] H. Moen, L.-M. Peltonen, J. Heimonen, A. Airola, T. Pahikkala, T. Salakoski, and S. Salanterä, “Comparison of automatic summarisation methods for clinical free text notes,” *Artificial intelligence in medicine*, vol. 67, pp. 25–37, 2016.

[23] R. Pivovarov and N. Elhadad, “Automated methods for the summarization of electronic health records,” *Journal of the American Medical Informatics Association*, vol. 22, no. 5, pp. 938–947, 2015.

[24] L. Plaza, A. Díaz, and P. Gervás, “A semantic graph-based approach to biomedical summarisation,” *Artificial intelligence in medicine*, vol. 53, no. 1, pp. 1–14, 2011.

[25] M. Moradi, “Cibs: A biomedical text summarizer using topic-based sentence clustering,” *Journal of biomedical informatics*, vol. 88, pp. 53–61, 2018.

[26] A. Sarker, Y.-C. Yang, M. A. Al-Garadi, and A. Abbas, “A light-weight text summarization system for fast access to medical evidence,” *Frontiers in Digital Health*, vol. 2, p. 45, 2020. doi: 10.3389/fdgth.2020.585559. [Online]. Available: https://www.frontiersin.org/article/10.3389/fdgth.2020.585559

[27] N. Rahman and B. Borah, “Improvement of query-based text summarization using word sense disambiguation,” *Complex & Intelligent Systems*, pp. 1–11, 2019.

DOI: 10.15439/2021F3
ISSN 2300-5963 ACSIS, Vol. 25

A Practical Solution to Handling Randomness and Imperfect Information in Monte Carlo Tree Search

Maciej Świechowski
QED Software, Warsaw, Poland
Email: maciej.swiechowski@qed.pl

Tomasz Tajmajer
Institute of Informatics
University of Warsaw, Poland
and *QED Software*, Warsaw, Poland

***Abstract*—This paper provides practical guidelines for developing strong AI agents based on the Monte Carlo Tree Search algorithm in a game with imperfect information and/or randomness. These guidelines are backed up by series of experiments carried out in the very popular game - Hearthstone. Despite the focus on Hearthstone, the paper is written with reusability and universal applications in mind. For MCTS algorithm, we introduced a few novel ideas such as complete elimination of the so-called nature moves, separation of decision and simulation states as well as a multi-layered transposition table. These have helped to create a strong Hearthstone agent.**

I. Introduction

Games of various kinds and forms have been challenging human minds since the ancient times [1]. Many games encode abstract problems, solving which requires high intelligence and well-developed reasoning ability. It is natural that people have started using them for more than just pure entertainment. With the inception of modern computers and artificial intelligence (AI), games have been often employed as testing environments [2], [3], [4], [5]. As of now, artificial intelligence in games is still a hot and growing research topic.

One of the recent trends in AI in games revolves around creating universal game-playing agents. Such an approach is believed to be closer to the roots of the AI and it is perceived as a step towards general intelligence. The most active research projects in this area are General Game Playing (GGP) [6], [7], [8] and General Video Game Playing (GVGP) [9]. Both projects include annual competitions open for the strongest programs. Both in GGP and GVGP, MCTS has become the state-of-the-art method. Since 2007, all winners of the GGP competition have used this algorithm. A particular reason for its success in general domains is the fact that MCTS requires only the rules of a game. Although it can take advantage of domain knowledge, as shown in [10], [11], it is fully operational without it.

Because MCTS is a statistics-based algorithm, it has helped to tackle non-deterministic hidden-information games, which had been particularly difficult for other tree search algorithms [12], [13], [14]. Nevertheless, MCTS is also used in games without non-determinism or hidden-information such as Go [15], Hex [16], Othello[17] or Havannah [18].

We describe a relatively complete approach to using MCTS in a game with hidden information and random effects [19], [20]. However, the specific game we have chosen for this study is Hearthstone: Heroes of Warcraft, developed by Blizzard Entertainment [21]. Hearthstone is an immensely popular game having around 70 million active players. It is a video card game that consists of series of duels. Players prepare 30-card decks of a chosen hero class. There are ten available hero classes. The goal is to use such a combination of minions, spells, weapons, secrets and alternative heroes to reduce the opponent's life total to zero. Minions can attack or be attacked. Spells usually deal damage, control the board (e.g., destroy minions), draw more cards or provide positive buffing/healing effects.

The paper is organized as follows. The next section is devoted to imperfect information, including Hearthstone-specific aspects of it (Section II-D), and the MCTS algorithm. Tree management using a multi-layered transposition table is shown in Section III. Section IV is devoted to the problem of dealing with randomness. All aspects presented in sections II to IV make up a complete MCTS-based agent. Based on the descriptions and pseudocode provided, the reader should be able to reproduce the agent and adapt it to a new game. The experiments and results for our MCTS-based agent in Hearthstone are presented in Section V. Finally, the last section is devoted to conclusions.

II. Imperfect Information

A. MCTS

This section is a short introduction to the Monte Carlo Tree Search algorithm (MCTS). We assume some familiarity of readers with the topic and recommend the survey [22] as a relatively exhaustive source of knowledge about MCTS.

The MCTS algorithm is the *state-of-the-art* method of performing the game tree search. The idea behind MCTS is to construct game tree iteratively, as depicted in Figure 1, by adding one node in each iteration. The algorithm uses simulations to gather statistical evidence about the quality of actions. The statistics include the average score, the number of visits to a node (state) and the number of times an action has been chosen in the iterations so far.

The aim of a selection policy is to maintain a proper balance between exploration (of not well-tested actions) and exploitation (of the best actions so far). The most common

This research was co-funded by the Smart Growth Operational Programme 2014-2020, financed by the European Regional Development Fund under a GameINN project POIR.01.02.00-00-0207/20, operated by The National Centre for Research and Development.

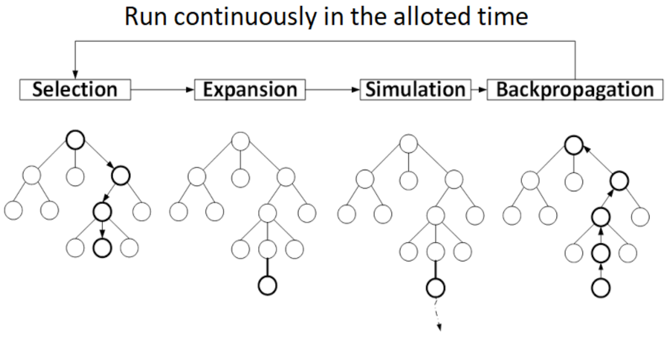


Fig. 1. The schema of the MCTS algorithm.

algorithm, which was also used for the experiments in this work, is called Upper Confidence Bounds applied for Trees (UCT) [23], [24].

B. The State-of-the-Art Approach

A game is of imperfect information if participating players cannot observe the complete state that affects the game. Even if some portion of the state is hidden only to certain players or only at certain moments, the game should still be considered as having imperfect information. Like randomness, imperfect information increases the combinatorial complexity of game tree search algorithms because the algorithms do not have access to the actual state (unless they cheat) and have to consider many potential states instead. There are two state-of-the-art methods of tackling hidden information with MCTS:

1) Perfect Information Monte Carlo Tree-Search [25] (PIMC): which performs determinization of hidden information and after that considers the game as a perfect-information one. Each determinization symbolizes a possible (parallel) world, in which a regular MCTS algorithm is executed. The standard PIMC algorithm performs many determinizations at the root level and combines statistics and decisions from them. The key problems with this approach are strategy fusion and nonlocality, both discussed in papers [25], [26]. The strategy fusion is manifested whenever an algorithm combines strategies determined for various worlds into a single optimal strategy. This can often lead to weak play, because the actual (unobservable) state is represented only by one of these worlds. Another effect of the strategy fusion with determinizations is manifested when an opponent is to make a partially observable move. After a determinization, each move is fully deterministic, so the PIMC algorithm can make different decisions based on a particular determinization of the opponent's partially observable move by the as discussed in work [26]. This is a problem of overfitting to specific determinizations. The nonlocality problem stems from the fact that determinizations might have different likelihoods of being accurate. In particular, some determinizations may be extremely unlikely, rendering their solutions irrelevant to the overall process.

2) Information Set Monte Carlo Tree Search [26] (ISMCTS): this algorithm introduces so-called information sets, which cluster states that are indistinguishable from a particular player's point of view. The ISMCTS algorithm greatly reduces the effects of strategy fusion and nonlocality, which are present in PIMC. However, ISMCTS requires a much more complex game model, which creates a huge implementation workload for developers. The authors of work [26] wrote that in ISMCTS, "*the player's choices of actions must be predicated on information sets, not on states*". In other words, the game simulator must allow for making both fully observable moves (for the sake of the regular game playing) and partially observable moves (for the sake of AI agents using ISMCTS). Working with Hearthstone, we have found out that this approach can be impractical for more complex games, especially video games with complex moves.

Another method, mentioned in studies [26], [27], is called Belief Distributions, which consists in modelling the decision process of players that have different access to information (typically, the mutual opponents). The history of observed actions forms an input for calculating the probability distributions of possible states. Such a method requires a good model, which is usually hand-crafted by experts. It has been applied in games such as Poker [28], in which one of the main aspects of the game is to guess what cards the opponent is holding.

C. Our Approach

Our solution combines determinizations and information sets. We introduce two distinct interfaces for representing the game state:

1) Game simulation state (GS-state): this interface allows performing all of the game's logic, such as determining legal moves, applying moves and updating states, checking if the game is in a terminal state, who won the game etc. It can only be used with complete information – either by some kind of game server / game-master that maintains the complete state (knows the correct one) or by a player after performing a determinization (guessing the state).

2) Information Set state (IS-state): this interface represents all information about the state of the game that is available to a particular player and is used by him or her to make decisions. The second property is very important - the information set may ignore, i.e., not contain, some information that is visible to the player if it is not that important from a decision process perspective. For example, let us assume that a health property of a player is a number from 0 to 100. The game AI designer may decide not to represent the health of a player as a numerical value, but rather a set of buckets $[0, 0], [1, 20], ..., [81, 100]$. Therefore, the maximum number of unique values of the health property is reduced from 101 to 6 in the IS-state. However, the GS-state has to operate on the maximum resolution to comply with the rules of the game properly. For the optimization of parameters chosen to be stored in information sets, machine learning algorithms can be used. Similarly, the IS-state may contain some redundant information from the GS-state point of view, e.g., whether a particular card has already been played in the game. The main consequence is that the proposed method allows for **complete separation of the state used for decisions (and gathering statistics of actions) and the state that is required for simulations**.

We believe that such a distinction has universal applicability and makes it easy to apply the ISMCTS algorithm in various games, no matter how the game simulator is written. In particular, the AI component based on the MCTS algorithm can be added after the game logic engine is written because the AI component will not put any constraints on the engine. This approach allows creating a game simulator separately from the AI module (a good software engineering practice), without making any sacrifices required for the AI. This property has been invaluable during development of the simulator for Hearthstone, especially in terms of how actions are represented and applied to a state. The IS-state is a static snapshot of transformed game state data, so there is no concept of applying actions to IS-states. The IS-state can be viewed as a so-called plain data object. As it will be shown in Section III, devoted to the storage of knowledge, the only functionality apart from storing data IS-state provides is the equality comparison. The relatively exhaustive pseudocode of the proposed approach is shown in Algorithm 1. The procedures which are not explained in detail in Algorithm 1 are discussed below:

- **updateRoot** - this procedure changes the root node in the tree when the current state of the game is changed. The procedure is explained in detail in Section III-A.
- **determinize** - this procedure guesses the hidden information in the game based on naive sampling among possible realizations with a uniform probability.
- **propagate** - the standard back-propagation phase of the MCTS.
- **simulate** - the standard simulation phase of the MCTS. This procedure starts with a given state, simulates till the end and gets players' scores.
- **tt.findOrCreate** - the procedure that finds a tree node that corresponds to the information set. It is explained in detail in Section III-B.

D. Information Sets in Hearthstone

We constructed the information sets in Hearthstone that consist of: (1) global publicly available information about the game such as the active player, game stage (e.g., mulligan or choose target), (2) data about Player1 and (3) data about Player2. The data about a player is modeled as a polymorphic structure with two possible types: the base *ISAnyPlayer* or *ISObservedPlayer* that inherits from *ISAnyPlayer*. The former type represents the perfect information portion of a player, i.e., everything that is visible about the player to all players. This includes such properties as the HP, armor, current crystals, maximum crystals, minions on board etc. However, to simplify the model and reduce the combinatorial complexity of states, we first sort minions by their ID numbers when doing the comparison of *IS-state* objects. As long as the two states contain the same set of minions (with the same attack values, health and ID numbers), their information sets are considered equal, even if positioning of the minions is different. The position of minions only matters when it affects attack or health. The ID is a number encoding a card's name, e.g., each "Prince Keleseth" card has the same ID. We also include a few specific properties of a player that could be theoretically derived from previous actions and states, such as whether a player has played an elemental card last turn (some cards gain extra effects based on this) or whether the "Prince Keleseth" buff has been applied in this game, which is a very unique effect that increases attributes of all minions in a player's deck.

Algorithm 1 The pseudocode of the proposed MCTS implementation.

```
procedure ITERATE($gs\_state$)
    $rootNode \leftarrow$ updateRoot($gs\_state$)
    $node \leftarrow rootNode$                                  ▷ current node
    while $elapsedTime < allotedTime$ do
        $gs\_movingState \leftarrow$ determinize($gs\_state$)
        GLOBALS::SELECTION ← RUNNING
        while GLOBALS::SELECTION is RUNNING do
            if $gs\_movingState.terminal \neq true$ then
                $node \leftarrow node.select(gs\_movingState)$
            end if
            propagate(simulate($gs\_movingState$))
        end while
    end while
end procedure

procedure NODE.SELECT($gs\_movingState$)
    $moves \leftarrow gs\_movingState.getMoves()$
    $curEdges \leftarrow []$
    for each $move$ in $moves$ do
        $edge \leftarrow allEdges[move]$
        if $edge$ not found then
            $edge \leftarrow$ new $edge(move)$
            $allEdges[move] \leftarrow edge$
        end if
        $edge.N \leftarrow +1$                    ▷ increment observed count
        $curEdges.push(edge)$
    end for
    $chosenEdge \leftarrow$ selection($curEdges$)    ▷ Using UCT formula
    $chosenMove \leftarrow chosenEdge.getMove()$
    $chosenEdge.V \leftarrow +1$                  ▷ increment visit count
    if $chosenEdge.V == 1$ then
        GLOBALS::SELECTION ← FINISHED
    end if
    $gs\_movingState.apply(chosenMove)$
    $is\_state \leftarrow$ createInformationSet($gs\_movingState$)
    $tt \leftarrow mcts.getTranspositionTable()$
    $chosenEdge.nextNode \leftarrow tt.findOrCreate(is\_state)$
    return $chosenEdge.nextNode$
end procedure
```

The *ISObservedPlayer* comes with additional data about the hidden information in the game: the hand and the secret zone. In an IS-state, we use a simpler model of a hand than the one used in a simulator. The hand is just a multi-set of cards' ID numbers, and any other properties of cards in the hand, such as effective mana cost, are ignored. The order of cards in hand is irrelevant. The secret zone is a set of secrets' ID numbers, because it is not possible to have more than one secret of a kind at the same time. Such a representation makes it possible to compare various approaches to modeling imperfect information:

1) Both players are modeled as *ISObservedPlayer* - this variant makes the assumption that the agent, which the tree is constructed for (the tree owner), determinizes the opponent and treats different determinizations as different states. This variant has the biggest granularity of states. **An exemplar consequence:** the sets of cards in both players' hands affect state comparisons.
2) Only the tree owner is modeled as *ISObservedPlayer* - here, we (i.e., the agent that is constructing the tree) do not differentiate between the states that differ with information we cannot see. **An exemplar consequence:** only the set of cards in the tree owner player's hand affects state comparison.
3) Only the currently active player is modeled as *ISObservedPlayer* - the idea is similar to (2), in (3), the tree owner hides the information about itself when simulating the opponent. This variant can be regarded as a symmetric version of (2). **An exemplar consequence:** the set of cards in a player's hands affects state comparison only if this player is to make a move in the state that is currently considered.

We have measured that both variants (2) and (3) work similarly without any significant difference in the playing strength of the resulting agent. The first variant, however, leads to significantly weaker bot because of the higher combinatorial complexity. In this variant, there are many more unique nodes in the transposition tables and therefore each node has less statistics. The strategy fusion problem arises with this approach as well.

III. Transposition Tables

Transposition tables [29], [30] were originally proposed as an enhancement to the alpha-beta algorithm, which reduces the size of minmax trees. As shown in paper [29], within the same computational budget, the enhanced algorithm significantly outperforms the basic one without the transposition tables. The term "transposition" refers to a state in the game that can be achieved in different ways.For simpler management, the MCTS tree is often modeled in such a way, that each unique sequence of actions leads to a state with a unique node in the tree. This leads to duplication of nodes, even for indistinguishable states. However, one of the benefits of such a duplication, is the fact that, whenever an actual action in the game is performed, the tree can be safely pruned into the sub-tree defined by the state the action lead to. All nodes that are either above the current one or on an alternative branch cannot be visited anymore, so there is no need to store them anymore. The problem is more complicated when transpositions are taken into account, and there is no longer one-to-one mapping between states and nodes. In such a case, the structure is no longer a tree per se, but a directed acyclic graph (DAG). When an action is played in the game, it is non-trivial to decide which nodes can be deallocated and which cannot because they might be visited again. In general, it would require a prediction model that can decide whether a particular state is possible to be encountered again. Storing all nodes, without deallocation, is detrimental not only for performance, but also the memory usage, which can become too high very quickly. Therefore, a worthwhile idea is to consider a probability a state will be encountered again and some threshold value based on which the nodes are pruned. Such an approach is suitable for a game-specific scenario but not for a universal case because there has not been proposed a general way of computing such a probability or setting the threshold. Therefore, we propose a solution based on reference counting described in the following subsection.

A. The Update Root Procedure

When an action is played in the actual game (not in a simulation), the algorithm first resets reference counts of all nodes. The node that corresponds to the state after the played action is the new root candidate. Next, the algorithm recursively traverses the nodes starting from the root candidate, increments the reference count in the currently visited node and proceeds through edges that lead to nodes with reference count equal to 0. After the recursive process terminates, all nodes with zero reference count are marked to be deallocated.

The above procedure is designed to be executed only once per action made in the game, so it does not bring high CPU overhead compared to the time required for simulations. We have measured this in Hearthstone. The profiling session consisted of 4 complete games with a one second clock for the moves. The average number of actions was 70 per game, what gives 280 executions of the update root procedure. The Monte Carlo simulations took 62% of the total time, whereas the update root procedure, described in the previous paragraph, took only 3.2% of the total time. However, because its only purpose is to reduce the memory usage, in the case of games, in which the memory footprint is low anyway, the procedure can be executed less frequently (e.g. every k actions or even once per game) to save some time. In Hearthstone, however, this procedure is required, because turning it off results in an out of memory exception after running for enough time (typically after a minute) with the setting of 20000 simulations (or more) per action.

B. Original Two-Layered Design

The proposed solution to storing the game tree is based on a two-layered structure of hash lookup tables (hashmaps). In many programming languages, the built-in hashmaps require keys to be integer values, so we decided to comply with this requirement. The whole idea is to find a node, if such exists, given an information set as the input. Our approach requires developers to implement two things. The first one is a boolean method *equals(otherIS)*, that checks whether two information sets are equal or not and returns *true* or *false*, respectively. Usually, the *equals()* method is used with the accompanying *hash()* method to speed up the retrieval of elements from a collection. Our approach is based on a more complex structure of two, preferably but not necessarily orthogonal, ways of computing hash values. Therefore, we introduce a two-dimensional hash function (or, equivalently, a pair of functions, one per dimension) that returns two values, each being an integer number, that represent distinct hashes of a state represented by an information set:

Let A and B be two information sets. The critical constraint on their two hash values is as follows:

$$(A = B) \Rightarrow (A.h^{'} = B.h^{'}) \wedge (A.h^{''} = B.h'') \qquad (1)$$

Such a two-level structure has been proposed based on analysis of complex games such as Hearthstone. For simple enough games, the hash function can return the same value in both dimensions, i.e., by setting $h^{'} = h^{''}$. In Hearthstone, any hashing function we tested that produced one number led to many collisions and therefore had a huge negative impact on the performance of searching nodes. Extending the hash into two dimensions allows for using independent measures without the need of dimension reduction and combining them within one equation. As a result, we managed to decrease the average time of finding nodes by an order of magnitude in comparison with a single-dimensional hash. The hash functions, we used in Hearthstone are as follows:

```
h' = ActivePlayerID +
    sum(for_each(P in Players):
    {
        4*P.MaxCrystals + 41*StageType +
        90*(P.HP + P.Armor +
        40*P.MinionsCount
        + 581*P.HandSize))
    }
h'' = sum(for_each(P in Players)
      {
         P.WeaponDurability +
         for_each(M in P.Minions):
            M.HP + 170*M.ID +
            21*(M.Attack+1)
      }
```

where $ActivePlayerID$ is the index of the player, encoded as 0 or 1 for Player 1 and Player 2, respectively; $StageType$ is $0, 1, 2, 3$, or 4 for *BasicAction*, *ChooseTarget*, *ChooseOne*, *Discover*, or *Mulligan*, respectively.

C. Using Transposition Tables in the Game

In our approach, transposition tables are used with the idea of separating the simulation game state and the decision game state. The decision game state is represented as an information set (IS-state). Firstly, we make a one-to-one mapping between information sets and nodes, so each IS-state has exactly one corresponding node. Information sets have already been defined in the previous sections as game state abstractions. A node is a functional structure that contains data required by the MCTS algorithm to operate. We managed to simplify nodes to only store a **collection of edges**. Because the MCTS version presented in this paper works with non-deterministic effects of actions, the edges cannot be stored as a fixed-sized array populated when a node is created. Instead, we store edges as a hashmap to satisfy the requirement that different sets of actions might be possible in consecutive visits by the MCTS algorithm to the same information set:

```
edges = hashmap<key: action, value: edge>
edge is composed of:
{
    stats: mcts_statistics
            (observedCount,
            visitCount,
            totalScore)
    current_next_node: node
}
```

Only the simulation game state is allowed to compute the legal moves, and it is possible that different simulation game states are mapped to the same information set. For example, there can be two simulation game states that differ with hidden information (indistinguishable from one player's perspective). Another example might be, when the AI designer/developer purposefully wants to simplify information sets and cluster more states together by ignoring some information available in the game. Therefore, we always look at the possible moves at the moment (by using the simulation game state) and then find the appropriate edge dynamically that corresponds to each specific move. An amortized cost of searching with hashmaps is $O(1)$. When a move is chosen, it is applied to the current GS-state and, based on the resulting GS-state, the IS-state (information set) is created. This information set is then used to retrieve the corresponding node.

IV. Randomness

Randomness is defined as a property of a game that some actions can have more than one outcome, i.e., that actions can lead to more than one distinct state, based on some arbitrary probability distributions. Players that participate in the game are not supposed to know the actual outcomes of the random effects before they materialize. They, however, may know the probability distributions underpinning random actions. In computer games, the game logic engine that is responsible for running the game (often referred to as the game server or game master) performs randomization in secrecy and informs the players when the effects of random actions become visible. A few examples of actions with nondeterministic outcomes are: shuffling a deck of cards, drawing a random card from a deck or rolling a die. Examples of games with randomness are Backgammon, Bridge, Poker, Settlers of Catan, Dungeons and Dragons, Magic the Gathering, and naturally, Hearthstone.

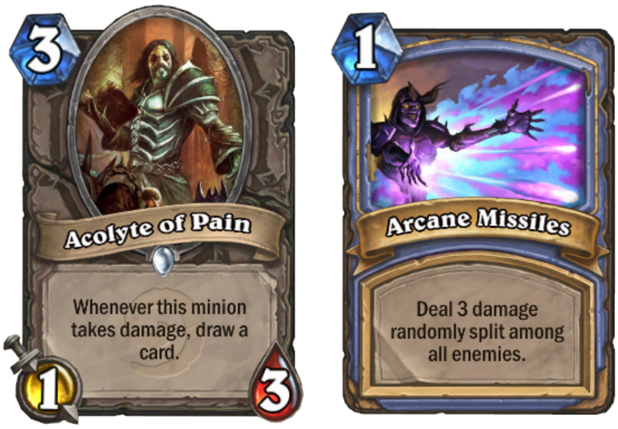

3
Acolyte of Pain
Whenever this minion takes damage, draw a card.
1
3
1
Arcane Missiles
Deal 3 damage randomly split among all enemies.

Fig. 2. Examples of cards including non-deterministic effects: drawing unknown cards (on the left) and random damage assignment (on the right).

Figure 2 shows two examples of cards with randomness in Hearthstone. Whether drawing a card is a random effect is a matter of interpretation and needs some further clarification. The rules of this game can be implemented in two ways. One way is to shuffle the deck of cards once and then each consecutive card is drawn from the top. In this case, drawing a card is not really a random effect but rather unveiling hidden information. The second way is not to shuffle the deck at all and give a random card to a player each time a card needs to be drawn. Both approaches are equivalent, because the deck is always supposed to be in a random order, and there are no effects in the game such as putting cards in a specific place in the deck.

Randomness raises the combinatorial complexity of a game. Let's consider a fully deterministic game with a branching factor of N. Now, if we make a change that each action on average has R possible random outcomes, the branching factor increases to N multiplied by R. Randomness is especially prevalent in card games. For instance, there are $52!$ ways a deck of 52 cards can be shuffled. Moreover, non-determinism of actions increases the difficulty of implementing the game engine to conform to the requirements of tree search algorithms such as MCTS. Consider the following problems:

- The MCTS algorithm stores statistics of players' actions: average score, total number of the action being observed, total number of the action being chosen. Let us now consider an action of playing the Arcane Missiles card shown in Fig. 2. In a given state, this action's average score should not be affected by how the damage will be split among the enemies, because the player could not know this outcome at the moment of playing the card. This suggests storing only one edge for this action. However, in the MCTS tree, there must be different nodes (states) that correspond to various outcomes of this action, which suggests having a separate edge for each result.
- The original Information Set Monte Carlo Tree Search algorithm requires the following property to be held. Let S be a state without perfect information. If identical sequences of actions $(a_1, ..., a_k)$ are applied from this state, then each one must end up with the same player active having the same set of available actions.

To tackle both mentioned problems, non-deterministic actions are split into a rational player's part and the so-called nature player's part. Continuing with the example of Arcane Missiles, the action would be split into:

```
PLAYER-1: Play Arcane Missiles
NATURE-P: Split 3 dmg among all enemies
```

or

```
PLAYER_1: Play Arcane Missiles
NATURE_P: Deal 1 dmg to a random enemy
NATURE_P: Deal 1 dmg to a random enemy
NATURE_P: Deal 1 dmg to a random enemy
```

The nature player is an artificial player that does not choose actions intentionally but rather performs them according to rules defined by the game (e.g., a probability distribution). The rational player observes such moves as random. A nature move determines the result of random calculations (it is one of the possible concrete realizations), therefore, after the move is generated, its outcome is deterministic. However, such a separation of player and nature moves as well as encoding determinations of random outcomes in the nature moves can be extremely difficult and time-consuming from the implementation point of view, that it might even be completely inapplicable in practice. In particular, video games are not designed in such a way to comply with the above mentioned model, so, if this model is used, one cannot necessarily separate the development of the actual game engine and the game AI.

Our goal was to make the development of the game engine maintainable for various games, so we decided to propose a model, in which actions can include any number of nondeterministic effects. The solution works in an integrated fashion with the information sets stored in transposition tables. Each time an action is made, the information set corresponding to the current state of simulation is searched for in the transposition table. Please refer to lines 34–37 in Algorithm 1 (pseudocode) for a possible implementation. Such an approach allows us not to introduce any explicit comparison operator for actions. In the proposed approach, two actions are different if they lead to different information sets, and they are equal otherwise. We keep only one edge for an action no matter how many resulting states it may have, so it represents a player's choice. The result of an action may vary in each MCTS iteration and because information sets and nodes correspond to each other, the same action might lead to different nodes in subsequent iterations. The UCT statistics gathered for the action are aggregated, so without any need for special treatment, the probability of random effects is taken into account. States which are more probable to occur will be visited more frequently by the MCTS, because it applies actions with their built-in randomness. At the same time, strong actions by means of the expected score (weighted by the probability of occurrence) will be chosen more often in the selection phase.

V. Results

A. Foreword

Before proceeding to the actual experiments results aimed at measuring efficiency of players, we wanted to first verify whether algorithms developed specifically to tackle the problems of randomness and incomplete information work in the game of Hearthstone.

The first test consisted in running 1,000,000 random simulations of the game. Each simulation had several unit tests to ensure the rules of Hearthstone are not violated. The second test consisted in running 100,000 matches between players that use the MCTS algorithm as defined in the paper. At this point, we were only interested whether there were no run-time errors at any point or problems with obtaining unequal states after applying the same sequence of actions from the same (stored and loaded) starting state. Both tests were successful. As a result, we can comment that that our way of implementing MCTS with dedicated algorithms for randomness and imperfect information is effectively error-free and does not put many constraints on how the simulator (forward-model) for a game is implemented. It is worthwhile noticing that our Hearthstone simulator was not specifically prepared to work with MCTS, yet the method is still easy to integrate with it.

B. Experimental setup

Our experimental setup consisted of the following AI players (controllers):

1) **Random** - an agent that performs actions according to the uniform random distribution among the currently available actions. The main purpose of including this agent is to make it serve as a baseline. A random controller can be also useful as a comparative benchmark in games, in which agents generally do not win against this agent 100% of the time; the relative difference of win-rates can be useful as a measure.
2) **EnhMCTS** - employs the MCTS algorithm as described in this paper. It is essentially a combination of the plain MCTS + our enhancement for randomness, incomplete information and transposition tables. We expected this player to consistently beat the *Random* player and not fall too much behind the "cheating" players described below.
3) **Hand Cheater** - is a *EnhMCTS* in which the player has perfect knowledge about the cards in the opponent's hand (hence, the cheater) but does not have access to the ordering of cards in decks.
4) **Full Cheater** - is a player that has perfect knowledge both about the opponent's hand and the ordering of cards in both players' decks. Such a player does not need to perform determinizations.

Both cheating controllers are used to find the upper bound on the performance of handling randomness and imperfect information.

The performance of players in Hearthstone is influenced by the decks they are using. For the experiments, we included four various types of decks that have also been used in the Hearthstone-based data mining competition [31]:

1) **Randomness** - this deck is made of cards with random effects exclusively. Examples of such cards are: *Arcane Missiles*, *Primordial Glyph* or *Animal Companion*. The idea of this deck was to increase the need of having a proper algorithm to handle randomness. This is also a deck that in theory gives the random player more chances to play better than more sophisticated approaches. The chosen hero for this deck was Mage.
2) **Minions** - this Hunter-based deck contains minion cards only. The goal was to have a relatively easy and streamlined deck to play.
3) **Control** - a "Cube Warlock" deck, which is very complex in terms of tactics. This deck allows agents to display their full potential as it requires long-term planning and tactical mastery.
4) **Aggro** - an average difficulty Paladin deck that is very fast and strong.

Each combination of decks and players was tested using 400 matches, which is above average in these kind of experiments.

C. Results with AI players

Let us start with baseline tests, which are depicted in Figure 3. Each grid corresponds to an experiment played between two players: *P1* and *P2*. The actual players used for the experiment are given in the top-most caption in each figure. Rows' labels are decks used by *P1* (the starting player), whereas columns are labeled by decks used by *P2*. The values in cells (i.e., on the intersections) contain the win ratio from the perspective of *P1*. A value of 1.0 means that *P1* has won all the games, a value of 0.5 represents a tie, and 0.0 is a perfect win for *P2*. If *P1* is equal to *P2*, then the main diagonal shows experiments played between identical players (so-called mirror matches); therefore, the results denote the first-player biases, i.e., how much more likely the player that goes first is to win in a particular setup.

The first test uses the weakest – Random – controllers. The main diagonal represents first move advantage, whereas other cells show biases between particular decks. For example, the *Aggro* deck has a clear advantage over the *Randomness* deck if it plays as first, whereas the *Minions* deck raises an upper hand against the *Control* when going first.

The next experiment, which is in the middle grid in Fig. 3, shows the same information for a pair of Full Cheaters, which are the strongest bots. It is interesting to note that these biases differ from those for the Random bots. The right-most part of Fig. 3 depicts those differences for each cell. In the experiment with Full Cheaters, $P1$ playing as *Aggro* significantly wins against any other deck. In comparison with Random vs. Random matches, Full Cheater is able to have a score higher by 0.20. The baseline results show that there

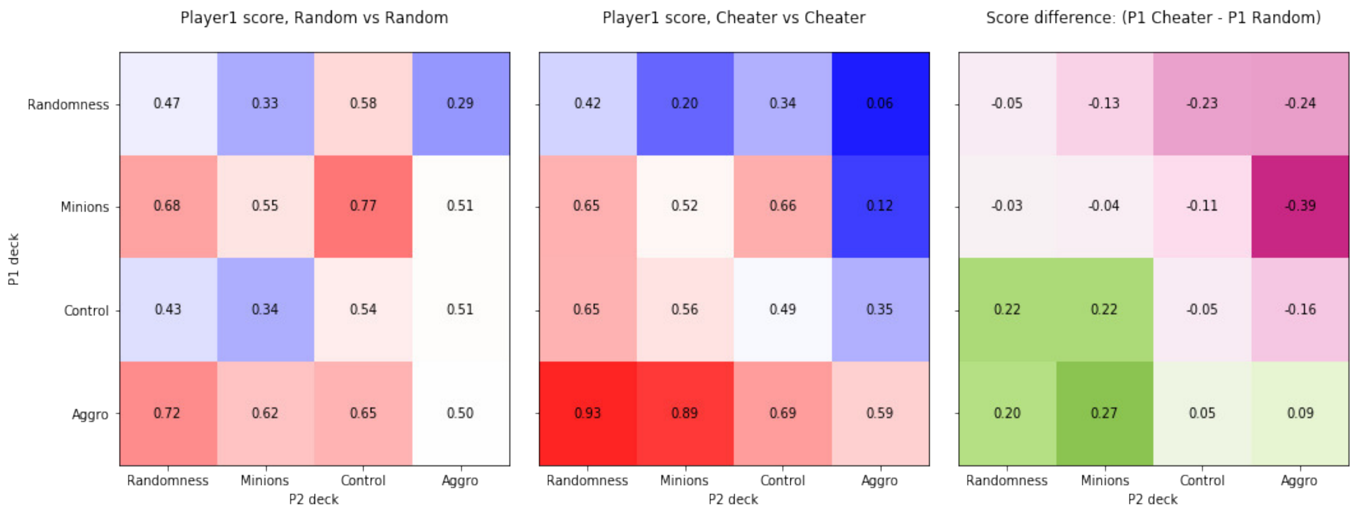


Fig. 3. The baseline scores. From the left: (1) Random Player vs Random Player, (2) Full Cheater vs Full Cheater and (3) the average score bias of Full Cheater compared to Random player

TABLE I
THIS TABLE CONTAINS THE AVERAGE SCORES OVER ALL MATCHES PLAYED BETWEEN THREE PAIRS PLAYERS THAT ARE DESCRIBED IN SECTION V-B. EACH TRIPLET OF COLUMNS IS DEVOTED TO ONE PARING OF PLAYERS. PLEASE NOTE THAT THE SCORES (P1 VS P2) AND (P2 VS. P1) ARE SHOWN FROM THE PERSPECTIVE OF THE PLAYER DENOTED BY P1.
THE THIRD COLUMN IN EACH TRIPLET DENOTES THE ADVANTAGE OF P2, WHICH IS CALCULATED ACCORDING TO THE FORMULA SHOWN IN EQUATION 2. THIS IS A DIFFERENCE IN TOTALSCORE(P2) - TOTALSCORE(P1).

		Hand Cheater (P1) vs. Full Cheater (P2)			**EnhMCTS (P1) vs Full Cheater (P2)**			**EnhMCTS (P1) vs Hand Cheater (P2)**		
		Score of P1		**Advantage**	**Score of P1**		**Advantage**	**Score of P1**		**Advantage**
P1 Deck	**P2 Deck**	**P1 vs P2**	**P2 vs P1**	**of P2**	**P1 vs P2**	**P2 vs P1**	**of P2**	**P1 vs P2**	**P2 vs P1**	**of P2**
Rnd	Rnd	0.35	0.58	0.07	0.39	0.54	0.07	0.47	0.53	0.00
Rnd	Minions	0.23	0.73	0.04	0.21	0.74	0.05	0.24	0.75	0.01
Rnd	Control	0.38	0.61	0.01	0.34	0.59	0.07	0.45	0.57	-0.02
Rnd	Aggro	0.05	0.94	0.01	0.07	0.91	0.02	0.08	0.93	-0.01
Minions	Rnd	0.62	0.4	-0.02	0.59	0.27	0.14	0.58	0.29	0.13
Minions	Minions	0.40	0.42	0.18	0.41	0.34	0.25	0.5	0.36	0.14
Minions	Control	0.62	0.26	0.12	0.57	0.17	0.26	0.73	0.19	0.08
Minions	Aggro	0.13	0.83	0.04	0.1	0.72	0.18	0.12	0.75	0.13
Control	Rnd	0.58	0.37	0.05	0.53	0.35	0.12	0.48	0.46	0.06
Control	Minions	0.48	0.39	0.13	0.39	0.34	0.27	0.42	0.46	0.12
Control	Control	0.45	0.49	0.06	0.38	0.34	0.28	0.46	0.36	0.18
Control	Aggro	0.24	0.6	0.16	0.24	0.47	0.29	0.28	0.54	0.18
Aggro	Rnd	0.94	0.06	0.00	0.93	0.04	0.03	0.94	0.06	0.00
Aggro	Minions	0.84	0.08	0.08	0.84	0.04	0.12	0.9	0.06	0.04
Aggro	Control	0.70	0.17	0.13	0.67	0.15	0.18	0.71	0.16	0.13
Aggro	Aggro	0.50	0.34	0.16	0.46	0.3	0.24	0.54	0.36	0.10
Column reference		I	II	III	IV	V	VI	VII	VIII	IX

are fundamental biases of decks and positions, which can be exploited by a good player – this is a general statement for Hearthstone.

One of the reasons we included the Random controller was to compare it to other agents. However, it turned out that each of our MCTS-based agents, i.e, Enhanced Vanilla MCTS, Hand Cheater and Full Cheater, achieves a 100% winrate over the random player. We can compare this result to 93% winrate reported in a literature [32], where the developed agent was also pitted against a (uniform) random player. This proves that **our implementation and setup of the MCTS algorithm itself is already very strong**.

Another experiment is aimed at measuring the impact of having full knowledge (Full Cheater), partial knowledge (Hand Cheater) or relying only on *fair* algorithms for incomplete knowledge (EnhMCTS). The baseline for the experiment is Full Cheater vs Full Cheater. All the results are presented in Table I. Please note that each experiment involves two players that switch sides after half of matches to avoid the starting role bias. However, to avoid confusion, the scores are always presented from the perspective of the first player, i.e., P1. The score of P2 can be calculated by $1 - Score(P1)$. Each third column contains an adjusted advantage in scores for P2. It is calculated as the total score of $P2$ minus the total score of P1. The total scores can be computed as follows:

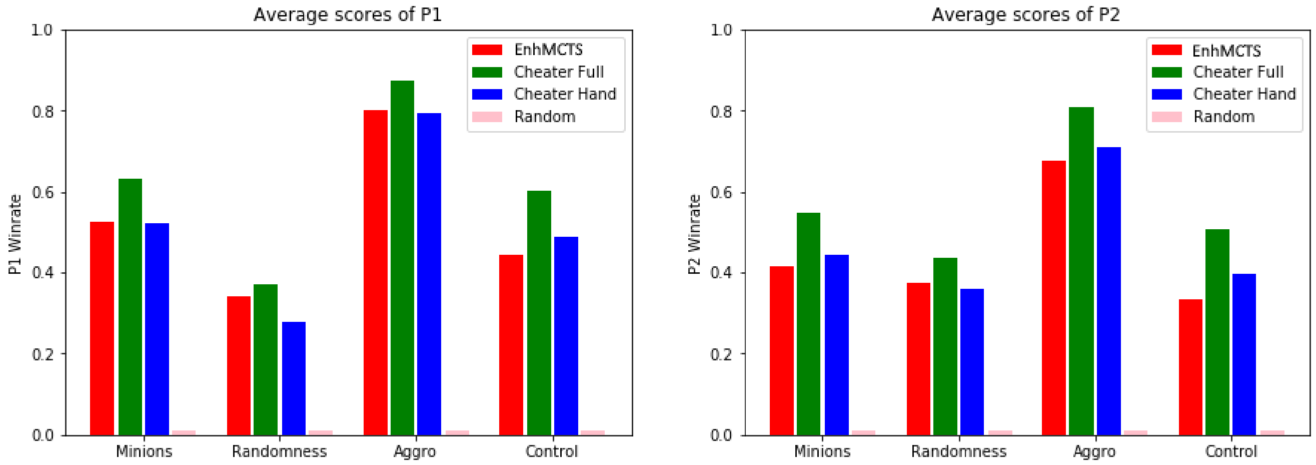


Fig. 4. Comparison of average win-rates for decks and bots. Scores are averaged for matches for any given deck-bot pair against all other decks and bots.

$$
\begin{aligned}
Score(P1) &= Score(\text{P1 vs P2}) + Score(\text{P2 vs P1})\\
Score(P2) &= 1 - Score(\text{P1 vs P2}) + 1 - Score(\text{P2 vs P1})
\end{aligned}
$$

The advantage of P2 expressed as a difference in scores reduces to:

$$
\begin{aligned}
&advantage(P2) = Score(P2) - Score(P1) =\\
&= 2 - 2 * Score(\text{P1 vs P2}) - 2 * Score(\text{P2 vs P1}) =\\
&= 1 - Score(\text{P1 vs P2}) - Score(\text{P2 vs P1})
\end{aligned}
\tag{2}
$$

Let us show that Hand Cheater is not much worse than Full Cheater. The results of this experiment are shown in columns I-III of Table I (please see the last row for the columns references). The third column shows the advantage of Full Cheater over Hand Cheater, and most of the values are not far from 0.00, which would denote equal performance. Only in three matchups is the total Full Cheater's score over 0.15 higher than the total score of Hand Cheater. In one of the 16 tested cases, i.e., *Minions* vs *Rnd*, Hand Cheater achieved better score.

Compared to the baseline scores in Fig 3, when Hand Cheater is playing as the first player, the biggest drop in performance is with *Minions* vs *Minions*, *Control* vs *Aggro* and *Aggro* vs *Aggro*. When Hand cheater is playing as the second player, the biggest drop in performance is for *Aggro* vs *Control*.

EnhMCTS is indeed a weaker player than Full Cheater, but the total score advantage of the latter was not greater than 0.29 in all cases. Please note that the theoretical maximum possible advantage is equal to 2.00, which would happen if one player wins all the games when playing both sides (the first and second to go). Such a case occurs when either of the three presented players – Full Cheater, Hand Cheater or EnhMCTS, faces the Random controller. With this in mind, the results achieved by EnhMCTS are promising.

When comparing the scores to the Cheater vs. Cheater baseline in Fig. 3, the worst cases for EnhMCTS playing as the first player are *Control* vs *Minions*, *Control* vs *Randomness* and *Aggro* vs *Aggro*. When EnhMCTS is the second to go, the worst cases are *Control* vs *Control*, *Control* vs *Aggro* and *Minions* vs *Aggro*.

Finally, EnhMCTS is slightly weaker than Hand Cheater as shown in column IX in Table I. It is worth noticing that, in overall, the advantage of Hand Cheater over EnhMCTS is similar to the advantage of Full Cheater over Hand Cheater.

Figure 4 shows the scores of each player averaged over each performed experiment using Random controller, EnhMCTS, Hand Cheater and Full Cheater with respect to the deck used. Unsurprisingly, Full Cheater is the strongest player. However, all of the MCTS-based players are relatively close to each other. The results show great potential of the methods for tackling incomplete information and randomness applied in EnhMCTS.

VI. Conclusions

The MCTS algorithm is the *state-of-the-art* method for searching the space of combinatorial games to find a good action to play in the current state. However, while the algorithm is clearly established for deterministic perfect-information games, it does not transfer directly onto games with randomness and imperfect information. Such games pose many challenges, and it is often unclear how to adapt MCTS for them. In this paper, we showed a very practical solution to this problem that is generic enough to be applied to any combinatorial game with randomness and incomplete information. The solution is based on three main pillars. The first one is a new approach to Information Set Monte Carlo Tree Search that operates on two levels of granularity. Information sets are used together with determinizations. The second pillar is dynamic resolution of randomness by matching states that result from random moves on-the-fly. The third pillar is a two-layered Transposition Table that is very fast to query and works particularly well with both the dynamic randomness resolution and information sets. Our approach does not enforce special constraints on how the game forward-model (simulator) is implemented, which is a huge advantage in a practical scenario, especially for commercial games.

To prove the method's efficacy, we have chosen the very complex game Hearthstone. The obtained results are very promising. Not only does the proposed algorithm work as intended, but also, as summarized in Figure 4, our methods for handling randomness and imperfect information fall only slightly behind a "cheating" agent that has full information about the game state.

For future work, we plan to perform further tests with different complex games with random effects and imperfect information. Next, we want to focus on the realm of non-card video-games, which feature huge combinatorial complexity and new challenges to overcome.

References

[1] L. Kurke, "Ancient Greek Board Games and How to Play Them," *Classical Philology*, vol. 94, no. 3, pp. 247–267, 1999.

[2] J. McCarthy, "Chess as the Drosophila of AI," in *Computers, chess, and cognition*. Springer, 1990, pp. 227–237.

[3] A. L. Samuel, "Some Studies in Machine Learning Using the Game of Checkers," *IBM Journal of research and development*, vol. 3, no. 3, pp. 210–229, 1959.

[4] C. E. Shannon, "XXII. Programming a Computer for Playing Chess," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 41, no. 314, pp. 256–275, 1950.

[5] M. Buro and T. Furtak, "RTS Games as Test-Bed for Real-Time AI Research," in *Proceedings of the 7th Joint Conference on Information Science (JCIS 2003)*, 2003, pp. 481–484.

[6] M. R. Genesereth, N. Love, and B. Pell, "General Game Playing: Overview of the AAAI Competition," *AI Magazine*, vol. 26, no. 2, pp. 62–72, 2005.

[7] M. Świechowski and J. Mańdziuk, "Self-Adaptation of Playing Strategies in General Game Playing," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 6, no. 4, pp. 367–381, Dec 2014.

[8] M. Świechowski, H. Park, J. Mańdziuk, and K.-J. Kim, "Recent Advances in General Game Playing," *The Scientific World Journal*, vol. 2015, 2015.

[9] J. Levine, C. B. Congdon, M. Ebner, G. Kendall, S. M. Lucas, R. Miikkulainen, T. Schaul, and T. Thompson, "General Video Game Playing," *Dagstuhl Follow-Ups*, vol. 6, 2013.

[10] S. Sharma, Z. Kobti, and S. Goodwin, "Knowledge Generation for Improving Simulations in UCT for General Game Playing," in *Australasian Joint Conference on Artificial Intelligence*. Springer, 2008, pp. 49–55.

[11] S. Haufe, D. Michulke, S. Schiffel, and M. Thielscher, "Knowledge-Based General Game Playing," *KI-Künstliche Intelligenz*, vol. 25, no. 1, pp. 25–33, 2011.

[12] I. Szita, G. Chaslot, and P. Spronck, "Monte-Carlo Tree Search in Settlers of Catan," in *Advances in Computer Games*. Springer, 2009, pp. 21–32.

[13] P. I. Cowling, C. D. Ward, and E. J. Powley, "Ensemble Determinization in Monte Carlo Tree Search for the Imperfect Information Card Game Magic: The Gathering," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 4, pp. 241–257, 2012.

[14] G. Van den Broeck, K. Driessens, and J. Ramon, "Monte-Carlo Tree Search in Poker Using Expected Reward Distributions," in *Asian Conference on Machine Learning*. Springer, 2009, pp. 367–381.

[15] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the Game of Go with Deep Neural Networks and Tree Search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[16] B. Arneson, R. B. Hayward, and P. Henderson, "Monte Carlo Tree Search in Hex," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 2, no. 4, pp. 251–258, 2010.

[17] D. Robles, P. Rohlfshagen, and S. M. Lucas, "Learning Non-Random Moves for Playing Othello: Improving Monte Carlo Tree Search," in *2011 IEEE Conference on Computational Intelligence and Games (CIG'11)*. IEEE, 2011, pp. 305–312.

[18] F. Teytaud and O. Teytaud, "Creating an Upper-Confidence-Tree Program for Havannah," in *Advances in Computer Games*. Springer, 2009, pp. 65–74.

[19] M. Swiechowski, T. Tajmajer, and A. Janusz, "Improving Hearthstone AI by Combining MCTS and Supervised Learning Algorithms," in *2018 IEEE Conference on Computational Intelligence and Games, CIG 2018, Maastricht, The Netherlands, August 14-17, 2018*, 2018, pp. 445–452. [Online]. Available: https://doi.org/10.1109/CIG.2018.8490368

[20] A. Janusz, T. Tajmajer, and M. Świechowski, "Helping AI to Play Hearthstone: AAIA'17 Data Mining Challenge," in *2017 Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 2017, pp. 121–125.

[21] A. K. Hoover, J. Togelius, S. Lee, and F. de Mesentier Silva, "The Many AI Challenges of Hearthstone," *KI-Künstliche Intelligenz*, vol. 34, no. 1, pp. 33–43, 2020.

[22] M. Świechowski, K. Godlewski, B. Sawicki, and J. Mańdziuk, "Monte Carlo Tree Search: A Review of Recent Modifications and Applications," 2021, submitted to Springer-Nature AI Reviews Journal. [Online]. Available: https://arxiv.org/abs/2103.04931

[23] L. Kocsis and C. Szepesvári, "Bandit Based Monte-Carlo Planning," in *Proceedings of the 17th European conference on Machine Learning*, ser. ECML'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 282–293.

[24] S. Gelly and Y. Wang, "Exploration Exploitation in Go: UCT for Monte-Carlo Go," in *NIPS: Neural Information Processing Systems Conference On-line trading of Exploration and Exploitation Workshop*, Canada, Dec. 2006. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00115330

[25] J. R. Long, N. R. Sturtevant, M. Buro, and T. Furtak, "Understanding the Success of Perfect Information Monte Carlo Sampling in Game Tree Search." in *AAAI*, 2010.

[26] P. I. Cowling, E. J. Powley, and D. Whitehouse, "Information Set Monte Carlo Tree Search," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 2, pp. 120–143, 2012.

[27] I. Frank and D. Basin, "Search in games with incomplete information: A case study using bridge card play," *Artificial Intelligence*, vol. 100, no. 1-2, pp. 87–123, 1998.

[28] M. J. Ponsen, G. Gerritsen, and G. Chaslot, "Integrating Opponent Models with Monte-Carlo Tree Search in Poker." in *Interactive Decision Theory and Game Theory*, 2010.

[29] A. Kishimoto and J. Schaeffer, "Transposition Table Driven Work Scheduling in Distributed Game-Tree Search," in *Proceedings of the 15th Conference of the Canadian Society for Computational Studies of Intelligence on Advances in Artificial Intelligence*, ser. AI '02. London, UK, UK: Springer-Verlag, 2002, pp. 56–68.

[30] J. Schaeffer, "The history heuristic and alpha-beta search enhancements in practice," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 11, pp. 1203–1212, 1989.

[31] A. Janusz, T. Tajmajer, M. Świechowski, Ł. Grad, J. Puczniewski, and D. Ślęzak, "Toward an Intelligent HS Deck Advisor: Lessons Learned from AAIA'18 Data Mining Competition," in *2018 Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 2018, pp. 189–192.

[32] D. Taralla, "Learning Artificial Intelligence in Large-scale Video Games: A First Case Study with Hearthstone: Heroes of Warcraft," Ph.D. dissertation, Université de Liège, Liège, Belgique, 2015.

DOI: 10.15439/2021F30
ISSN 2300-5963 ACSIS, Vol. 25

Automated creation of parallel Bible corpora with cross-lingual semantic concordance

Jens Dörpinghaus*, Carsten Düing†
* University of Pretoria, Faculty of Theology and Religion, Hatfield, Pretoria, South Africa,
Email: u21829927@tuks.co.za
† Carsten Düing has been with the Faculty for Mathematics and Informatics,
Fernuniversität Hagen, Germany at the time of this work

***Abstract*—Here we present a novel approach for automated creation of parallel New Testament corpora with cross-lingual semantic concordance based on Strong's numbers. As scientific editions and translations of Bible texts are often not free to use for scientific purposes and are rarely free to use, and due to the fact that the annotation, curation and quality control of alignments between these texts are quite expensive, there is a lack of available Biblical resources for scholars. We present two approaches to tackle the problem, a dictionary-based approach and a Conditional Random Field (CRF) model and a detailed evaluation on annotated and non-annotated translations. We discuss a proof-of-concept based on English and German New Testament translations. The results presented in this paper are novel and according to our knowledge unique. They present promising performance, although further research is necessary.**

I. Introduction

Building a concordance of texts, automated text alignment and automated text translations are well studied topics in research. A *semantic concordance* is a widely used approach to link text corpora with data and values in lexicons, see [1]. Even in the humanities a lot of research has been done within this wide field of text mining and automated text processing. Coming to the field of what some call *Digital Theology* as a subfield of *Digital Humanities* and its intersection to ancient languages we still see a lot of challenges, although the problems themselves may seem easy and a standard task.

Here, we want to tackle the challenge of automated annotations of words within New Testament texts to create parallel Bible corpora in different languages. Thus, our goal is to create a cross-lingual concordance alignment for New Testament texts and translations. These are widely used for research and teaching. Our approach is restricted to the mapping between original and translated words given both the translation with or without further information and the Greek source with morphological information.

Research on Biblical texts and translations of course has a long tradition and translations have been widely used.There was a great increase in the amount of different Bible translations in the nineteenth century and thus, the research in this field increased also, see [2]

New approaches from computer science have also been used to evaluate translations and texts but only really took off in the last 30 years as it became more accessible to scholars with a different background. It is possible to use these methods to understand the manual curation and understanding of text and it would be to improve the technological solutions for automated approaches. Here, Clivaz states in 2017 [3] that only very little research has been done in this field and Anderson underlines the theologians lack of interest for digital and modern text mining methods a year later [4]Only the fields of digital manuscripts, Digital Academic Research and Publishing show some progress [5]. This work tries to be a first step to close this gap.

As scientific editions and translations of Bible texts are often not free to use and due to the fact that working on them is quite expensive, there is a lack of available Biblical resources for scholars. The aim of this work is to develop and evaluate novel approaches for automated generation of alignment for parallel Bibles leading to cross-lingual semantic concordance.

This paper is divided into six sections. The first introduces the problem. The second section gives a brief overview over the state of the art and related work. The third section is dedicated to the data foundation. We will also discuss the annotation style and the selection of training and test data. In the fourth section, we present two approaches to tackle the problem. We introduce a dictionary-based approach and a CRF model. The fifth section is dedicated to experimental results on annotated and non-annotated translations. Our conclusions are drawn in the final section. The results presented in this paper are novel and according to our knowledge unique. They present promising performance, although further research is necessary.

II. Related Work

Since only little research has been done in this field, we list all material available even if there tasks are only tangentially related. In Biblical research *The Exhaustive Concordance of the Bible* from 1890 is widely used to link words from Biblical texts to dictionary entries. These so-called Strong's numbers can be used to create automatic aligned parallel texts, see [6] or [7] who created semantic maps from parallel text data. Here, texts in multiple languages are presented together [8]. Although a lot of approaches are based on machine translation in Biblical research, these texts are still mainly hand-crafted, e.g. [9] or [10].Even if the Bible is often used as training model or reference model for unsupervised learning models for translation, see for example [11], [12] or [13], only few approaches have also been made to analyze religious or theological texts with methods from AI and text mining.

To cover the language related question other scholars examined the impact of computer technologies on Bible translations and discussed their limitations [14]. Bible translations usually not being in the scope of linguistic research, but interesting for the history of language, there is a wide range of publications and analyses of recent translations, see e.g. [15] and [16]. There is also a considerable amount of literature on Bible translations [17]. It is important to notice that Bible translation is not only about decisions between translation strategies like formal or dynamic equivalence.

Encoding linguistic information in multi-language documents produces *Interlinear Glossed Text* (IGT). Biblical texts are usually well-studied and thus both references to the Strong's numbers as well as morphological information are available for Hebrew and Greek texts. Automated glossing is also a widely studied field, see [18] or [19]. These approaches have never been used to create interlinear glossed Biblical texts. Only some little research has been done on the Qur'an [20]. For automated translations, there are no resources available for ancient Greek [21]. Other approaches, like GASC [22], build a Bayesian model describing evolution of words and meanings in ancient texts. They state "a lack of previous works that focussed on ancient languages". Thus, not only the target texts form a new field, but we can also only build upon very little work within the field of automated translations.

III. Data

Here, we will focus on Greek original text, German and English Bible translations, although this approach can be used for any other language. There are several software packages available to access Biblical texts. Some commercial software like Logos offer no or only very limited access to their API[1]. Thus, we did our work on the basis of the SWORD Project, which offers a full API available under GNU license[2]. As a basis for the Greek text we used the SBLGNT 2.0 from Tyndale House, based on SBLGNT v.1.3 from Crosswire. This text is with some minor changes comparable to the Nestle-Aland/United Bible Societies text. The English texts are based on KJV (1769, King James Version), ASV (1901 American Standard Version) and ESV (English Standard Version, 2011). The dictionaries are based on the original Strong's Dictionaries or are extracted from the texts. The German texts are based on Luther (1912), Leonberger Bibel (2017), the Greek-German dictionary by Gerhard Kautz and for a detailed analysis on some excerpts of newer translations. Beside of them, all data is available with a free license. See http://www.crosswire.org/sword/modules/ for details of these packages.

Different approaches for translating Biblical texts exists. KJV, ESV and ASV follow a traditional word-for-word approach, also known as formal equivalence. The Leonberger Bibel follows the same approach, whereas Luther 1912 also has elements from the thought-for-thought approach known as dynamic equivalence. For testing purposes we will also consider translations which use a paraphrase approach. For a detailed overview about Bible translations see [2].

A. Annotation Style

There are several annotations which can be displayed in different ways. Here, we rely on the HTML-output. Both lemma and morphology information are included in w-tags. For example in Acts 1:1:

```
<w lemma="strong:G3303" morph="
    robinson:PRT" savlm="strong:G3303" src
    ="2"/>
```

We will use this annotation style both for extracting information, storing and comparing them.

B. Training and Test Data

To collect the training data, we can use the complete New Testament texts mentioned above. This leads to 7,957 verses in each version. There are 5,624 entries in the Strong's dictionary. We tested our models both on a random subset of the same and other translations. In addition, we will test our model on some verses from newer versions, e.g. the recent German Luther-Bible. Here, the verses are curated by hand.

IV. Methodology

A. Modeling

Here, we have Biblical texts containing verses. Each verse X contains a sequence of words, thus $X = x_1, ..., x_N$. Given two languages L and L' we have two sequences

$$\begin{aligned} X^L &= x_1^L, ..., x_N^L \\ X^{L'} &= x_1^{L'}, ..., x_M^{L'} \end{aligned}$$

And we want to model the target glossing $f : X^L \rightarrow X^{L'}$ that contains mappings from a word origin $x_i^L \in X^L$ to another word $x_j^{L'} \in X^{L'}$. Let Y be a sequence of all mappings, than we need to compute $P(Y|X^L)$.

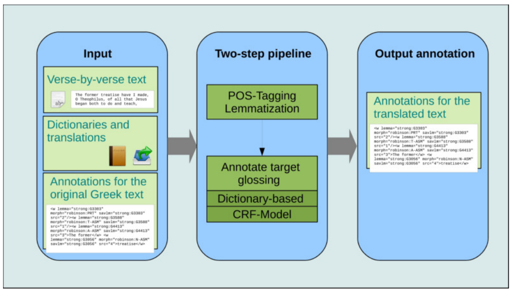

Input
Verse-by-verse text
The former treatise have I made, O Theophilus, of all that Jesus began both to do and teach,
Dictionaries and translations
Annotations for the original Greek text
<w lemma="strong:G3303" morph="robinson:PRT" savlm="strong:G3303" src="2"/><w lemma="strong:G3588" morph="robinson:T-ASM" savlm="strong:G3588" src="1"/><w lemma="strong:G4413" morph="robinson:A-ASM" savlm="strong:G4413" src="3">The former</w> <w lemma="strong:G3056" morph="robinson:N-ASM" savlm="strong:G3056" src="4">treatise</w>
Two-step pipeline
POS-Tagging Lemmatization
Annotate target glossing
Dictionary-based
CRF-Model
Output annotation
Annotations for the translated text
<w lemma="strong:G3303" morph="robinson:PRT" savlm="strong:G3303" src="2"/><w lemma="strong:G3588" morph="robinson:T-ASM" savlm="strong:G3588" src="1"/><w lemma="strong:G4413" morph="robinson:A-ASM" savlm="strong:G4413" src="3">The former</w> <w lemma="strong:G3056" morph="robinson:N-ASM" savlm="strong:G3056" src="4">treatise</w>

Figure 1. The proposed two-step method. First, we use a POS-Tagging and Lemmatization to extract the word to be matched. Then, we annotate the target glossing, either with the dictionary-based method or using a CRF-Model. As input, we use the target text (a translated text) verse-by-verse, the original Greek text containing the annotations and some additional information from dictionaries and Biblical translations.

[1] See for example https://wiki.logos.com/Logos_4_COM_API.

[2] See http://crosswire.org/sword/index.jsp

Algorithm 1 DICTIONARY-BASED-MATCHES

```
Require: Sequences of words $X^L = x_1^L, ..., x_N^L$ with dictionary mapping to $d(x_i^L)$ to dictionary $D$ and in target language $X^{L'} = x_1^{L'}, ..., x_M^{L'}$.
Ensure: Mapping $f : X^L \to X^{L'}$.
    for $c$ in $POS$: do
        for $x_i^L$ in $c$: do
            find $x_j^{L'}$ with $\min \delta(lem(d(x_i^L)), lem(x_j^{L'}))$
            assign $f(x_i^L) = x_j^{L'}$
        end for
    end for
    return $f$
```

Here, we propose a two-step method. As input we use the target text (a translated text) verse-by-verse, if needed, the original Greek text containing the Strong's annotations and some additional information from dictionaries and Biblical translations. First, we use POS-tagging and lemmatization to extract the word to be matched. Then, we annotate the target glossing, either with the dictionary-based method which is a natural fit because a lot of features are available or using a CRF-Model which is one of the standard solutions in current NLP. See figure 1 for an illustration.

B. Dictionary-based approach

After detecting parts-of-speech in the target text, we can sort words from the original Greek text and the target language. This helps to reduce the target set of words. Since we know the Greek Strong's numbers, we can use lemmatization to compare words and assign the best fit, see algorithms 1.

We need to choose a proper distance function δ (like Levenshtein distance or cosine similarity) and we need to choose proper dictionaries.

By language, we can either rely on dictionaries: the Greek-English dictionary by Dr. Ulrik Sandborg-Petersen and the Greek-German dictionary by Gerhard Kautz, both released under CC license. In addition, we build dictionaries from the annotated Biblical texts presented in section III. Here, we for every Strong's number we collected words in a particular Bible translation.

In order to make this data available, we wrote an importer to create a list of words in the target language associated with a Strong's number. This dictionary-based approach is a lazy learner approach, since first we learn dictionaries but the comparison and assignment is done in a separate step. Thus, we will now introduce a different approach using CRF models.

C. CRF-Model

Our second approach uses a linear-chain Conditional Random Field (CRF), see [23]. Here, we train a sequence model where the input consists of words and the output of a Strong's-labels. [18] used this method to automatise the gloss generation in interlinear glossed texts. Here, we used sklearn-crfsuite v0.3.6 to build the CRF models. For training, we used stochastic gradient descent with L2 regularization and a maximum of 50 iterations.

Target	Luther1912			GerLeoNA28		
Source	P	R	F1	P	R	F1
Luther1912	.75	1	.84	.45	.83	.55
GerLeoNA28	.69	.96	.78	.56	.95	.67
Luther1912 + GerLeoNA28	.77	1	.86	.57	.92	.67
CRF Luther1912	.12	.14	.13	-	-	-
CRF GerLeoNA28	-	-	-	.53	.52	.52

Table I

EVALUATION OF DIFFERENT GERMAN TARGET TRANSLATIONS. THE BASIS ARE EITHER A COMBINATION OF DICTIONARY-BASED APPROACHES OR THE CRF MODEL. HERE, P REFERS TO PRECISION, R TO RECALL, F1 TO F1-SCORE.

Target	KJV			ASV			ESV		
Source	P	R	F1	P	R	F1	P	R	F1
KJV	.50	.83	.58	.69	1	0.79	.63	.96	.74
ASV	.44	.79	.53	.73	.96	.81	.66	.96	.75
ESV	.38	.71	.46	.72	.96	.80	.67	.96	.78
ASV + ESV	.38	.71	.46	.72	.96	.80	.68	.96	.78
KJV + ASV + ESV	.41	.75	.49	.71	.96	.79	.68	.96	.78
CRF KJV	.26	.20	.20	-	-	-	-	-	-
CRF ASV	-	-	-	.33	.33	.33	-	-	-
CRF ESV	-	-	-	-	-	-	.27	.25	.25

Table II

EVALUATION OF DIFFERENT ENGLISH TARGET TRANSLATIONS. THE BASIS ARE EITHER A COMBINATION OF DICTIONARY-BASED APPROACHES OR THE CRF MODEL. HERE, P REFERS TO PRECISION, R TO RECALL, F1 TO F1-SCORE.

For our testing purpose, we include basic linguistic features, the source and previous and following words. For training purposes, we can again rely on the Biblical texts which are already annotated with Strong's numbers.

D. Evaluation

The performance of each approach is evaluated by comparing each annotation in each final output to the test data set provided from annotated Biblical texts. Thus, we need to cross-evaluate different input scenarios against different and similar output scenarios. The Greek-English dictionary by Dr. Ulrik Sandborg-Petersen and the Greek-German dictionary by Gerhard Kautz were not presented, because it was not possible to extract the exact proposed translations with reasonable effort.

Since our approach produces Strong's numbers annotations for words in translated text, the first question is if this leads to proper assignments on the *same* text. We will also evaluate, if combining different models will lead to better solutions. Because these approaches may predict Strong's numbers that have more or fewer occurrences in the text we add both precision and recall to our evaluation. These metrics are presented as a micro-average value over all verses.

Further, we will analyze how these systems will work on unanotated translations. For this purpose, a few verses have been chosen to evaluate the output.

V. RESULTS

A detailed evaluation with to precision, recall, and F1-Score can be found in tables I for German translations and II for

English translations. These tests showed unexpectedly that the CRF models were not competitive with the dictionary-based approaches. We will discuss this observation and possible reasons later.

The dictionary based approaches on German translations (table I) show very promising results. The recall value is really high, although the precision value increases, if more dictionaries are combined. Although we can see a different behavior for Luther1912 and GerLeoNA28. For the latter, the combination of both dictionaries increases the precision, but also decreases recall value. It is crucial to note that a combination of dictionaries needs a careful investigation. Here, one of the reasons might be that although both translation have been done with the same approach, there is more than hundred years in between them. So the words and their meanings might have changed. In the next section, we will do some preliminary observation on more recent translations.

This is even more significant for the evaluation of English translations in table II. ESV and ASV are both bases on KJV and again there are more than hundred years in between (1769, 1901, 2011). The two most recent translations show a good result, the recall value is high and the precision value increases with the matching dictionary. The most remarkable can be found when using KJV for a combination of dictionaries, it decreases the values significantly. This result has further strengthened our confidence that it is crucial to evaluate the dictionary basis for this approach.

VI. Conclusion and Future Work

This paper has described a first approach to automated annotations of words within New Testament texts to create parallel bible corpora in different languages to create a cross-lingual concordance alignment for New Testament texts and translations. We proposed a lazy-learner and an eager-learner approach: A dictionary-based and a CRF-based approach.

While the amount of training data was due to strict license politics in the field of Theology relatively low, we could nevertheless get promising results for some translations. This method can't be applied to translations following a paraphrase approach. This will hopefully lead to further research and a better understanding of special requirements within the field of theology and in particular ancient languages. Here, we see the need for more models and methods since there are no resources available for ancient Greek.

Our analysis of errors reveals a number of questions and also possible further improvement. First, we need to consider if more translations and Biblical texts can be used as training data. Although not in every case the results could be improved when more dictionaries were used a better data foundation together with improvements in modeling and algorithms will improve the results. Second, we need to investigate why some parts of speech, in particular nouns and conjunctions, do not work well at all. Finally, we need to make an in-depth error analysis why the CRF models do not work as expected. Here, we will invest weather a better feature selection (for example POS tagging or dependency labels) will improve the results.

While our proof of concept is both working and generic it is still very early work on a problem which needs more attention. We hope that it will also highlight the importance of more interdisciplinary research in this field.

References

[1] S. Landes, C. Leacock, and R. I. Tengi, "Building semantic concordances," *WordNet: An electronic lexical database*, vol. 199, no. 216, pp. 199–216, 1998.

[2] B. Metzger, *The Bible in Translation: Ancient and English Versions*, ser. Biblical studies. Baker Publishing Group, 2001.

[3] C. Clivaz, "Die bibel im digitalen zeitalter: Multimodale schriften in gemeinschaften," *Zeitschrift für Neues Testament*, vol. 20, no. 39/40, pp. 35–57, 2017.

[4] C. Anderson, "Digital humanities and the future of theology," 2018.

[5] C. Clivaz, A. Gregory, and D. Hamidović, *Digital Humanities in Biblical, Early Jewish and Early Christian Studies*. Brill, 2013.

[6] M. Cysouw, C. Biemann, and M. Ongyerth, "Using strong's numbers in the bible to test an automatic alignment of parallel texts," *STUF-language typology and universals*, vol. 60, no. 2, pp. 158–171, 2007.

[7] B. Wälchli, "Similarity semantics and building probabilistic semantic maps from parallel texts," *Linguistic Discovery*, vol. 8, no. 1, pp. 331–371, 2010.

[8] M. Simard, "Building and using parallel text for translation," *The Routledge Handbook of Translation and Technology*, pp. 78–90, 2020.

[9] A. Yli-Jyrä, J. Purhonen, M. Liljeqvist, A. Antturi, P. Nieminen, K. M. Räntilä, and V. Luoto, "Helfi: a hebrew-greek-finnish parallel bible corpus with cross-lingual morpheme alignment," *arXiv preprint arXiv:2003.07456*, 2020.

[10] N. Rees and J. Riding, "Automatic concordance creation for texts in any language," *Proceedings of Translation and the Computer*, vol. 31, 2009.

[11] M. Diab and S. Finch, "A statistical word-level translation model for comparable corpora," MARYLAND UNIV COLLEGE PARK INST FOR ADVANCED COMPUTER STUDIES, Tech. Rep., 2000.

[12] P. Resnik, M. B. Olsen, and M. Diab, "The bible as a parallel corpus: Annotating the 'book of 2000 tongues'," *Computers and the Humanities*, vol. 33, no. 1, pp. 129–153, 1999.

[13] C. Christodouloupoulos and M. Steedman, "A massively parallel corpus: the bible in 100 languages," *Language resources and evaluation*, vol. 49, no. 2, pp. 375–395, 2015.

[14] J. D. Riding, "Statistical glossing, language independent analysis in bible translation," *Translating and the Computer*, vol. 30, 2008.

[15] J. Renkema and C. van Wijk, "Converting the words of god: An experimental evaluation of stylistic choices in the new dutch bible translation," *Linguistica Antverpiensia, New Series–Themes in Translation Studies*, no. 1, 2002.

[16] L. De Vries, "Bible translation and primary orality," *The Bible Translator*, vol. 51, no. 1, pp. 101–114, 2000.

[17] G. G. Scorgie, M. L. Strauss, S. M. Voth *et al.*, *The challenge of Bible translation: Communicating God's Word to the world*. Zondervan Academic, 2009.

[18] A. McMillan-Major, "Automating gloss generation in interlinear glossed text," *Proceedings of the Society for Computation in Linguistics*, vol. 3, no. 1, pp. 338–349, 2020.

[19] X. Zhao, S. Ozaki, A. Anastasopoulos, G. Neubig, and L. Levin, "Automatic interlinear glossing for under-resourced languages leveraging translations," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 5397–5408.

[20] A. B. Muhammad, *Annotation of conceptual co-reference and text mining the Qur'an*. University of Leeds, 2012.

[21] E. Biagetti, C. Zanchi, and W. M. Short, "Toward the creation of wordnets for ancient indo-european languages," in *Proceedings of the 11th Global Wordnet Conference*, 2021, pp. 258–266.

[22] V. Perrone, M. Palma, S. Hengchen, A. Vatri, J. Q. Smith, and B. McGillivray, "GASC: Genre-aware semantic change for Ancient Greek," in *Proceedings of the 1st International Workshop on Computational Approaches to Historical Language Change*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 56–66. [Online]. Available: https://www.aclweb.org/anthology/W19-4707

[23] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," *Proceedings of the Eighteenth International Conferenceon Machine Learning*, 2001.

DOI: 10.15439/2021F7
ISSN 2300-5963 ACSIS, Vol. 25

Evaluation of Neural Network Transformer Models for Named-Entity Recognition on Low-Resourced Languages

Ridewaan Hanslo
University of Pretoria
Gauteng, South Africa
Email: ridewaan.hanslo@up.ac.za

***Abstract*—Neural Network (NN) models produce state-of-the-art results for natural language processing tasks. Further, NN models are used for sequence tagging tasks on low-resourced languages with good results. However, the findings are not consistent for all low-resourced languages, and many of these languages have not been sufficiently evaluated. Therefore, in this paper, transformer NN models are used to evaluate named-entity recognition for ten low-resourced South African languages. Further, these transformer models are compared to other NN models and a Conditional Random Fields (CRF) Machine Learning (ML) model. The findings show that the transformer models have the highest F-scores with more than a 5% performance difference from the other models. However, the CRF ML model has the highest average F-score. The transformer model's greater parallelization allows low-resourced languages to be trained and tested with less effort and resource costs. This makes transformer models viable for low-resourced languages. Future research could improve upon these findings by implementing a linear-complexity recurrent transformer variant.**

I. Introduction

XLM-Roberta (XLM-R) is a recent transformer model that has reported state-of-the-art results for Natural Language Processing (NLP) tasks and applications, such as Named-Entity Recognition (NER), Part-of-Speech (POS) tagging, phrase chunking, and Machine Translation (MT) [2], [8]. The NER and POS sequence tagging tasks have been extensively researched [1]-[6], [8], [9]. However, within the past few years, the introduction of new Deep Learning (DL) transformer model architectures such as XLM-R, Multilingual Bidirectional Encoder Representations from Transformers (M-BERT) and Cross-Lingual Language Model (XLM) lowers the time needed to train large datasets through greater parallelization [7]. This allows low-resourced languages to be trained and tested with less effort and resource costs, with state-of-the-art results for sequence tagging tasks [1], [2], [8]. M-BERT as a single language model pre-trained from monolingual corpora performs very well with cross-lingual generalization [10]. Furthermore, M-BERT is capable of capturing multilingual representations [10]. On the other hand, XLM pre-training has led to strong improvements on NLP benchmarks [11]. Additionally, XLM models have contributed to significant improvements in NLP studies involving low-resource languages [11]. These transformer models are usually trained on very large corpora with datasets in terabyte (TB) sizes.

A recent study by [1] researched the "*Viability of Neural Networks for Core Technologies for Resource-Scarce Languages*". These resource-scarce languages are ten of the 11 official South African (SA) languages, with English being excluded. The languages are considered low-resourced, with Afrikaans (af) being the more resourced of the ten [1], [9]. This recent study looked at sequence tagging (POS tagging and NER) and sequence translation (Lemmatization and Compound Analysis), comparing two Bidirectional Long Short-Term Memory with Auxiliary Loss (bi-LSTM-aux) NN models to a baseline Conditional Random Fields (CRF) model. The annotated data used for the experiments are derived from the National Centre for Human Language Technology (NCHLT) text project. The results suggest that NN architectures such as bi-LSTM-aux are viable for NER and POS tagging tasks for most SA languages [1]. However, within the study by [1], NN did not outperform the CRF Machine Learning (ML) model. Rather they advised further studies be conducted using NN transformer models on resource-scarce SA languages. For this reason, this study builds upon the previous study, using the XLM-R DL architecture. Therefore, the purpose of this study is to evaluate the performance of the NLP NER sequential task using two XLM-R transformer models. In addition, the experiment results are compared to previous research findings.

A. Research Questions

RQ_1 – How does the XLM-R neural network transformer models perform with NER on the low-resourced SA languages using annotated data?

RQ_2 – How does the XLM-R transformer models compare to other neural network and machine learning models with NER on the low-resourced SA languages using annotated data?

B. Paper Layout

The remainder of this paper comprises of the following sections: Sect. II provides information on the languages and datasets; Sect. III presents the language model architecture. The experiment settings are presented in Sect. IV and the results and a discussion of the research findings are provided in Sect. V. Section VI concludes the paper with the limitations of this study and recommendations for further research.

II. Languages and Datasets

As mentioned by [1], SA is a country with at least 35 spoken languages. Of those languages, 11 are granted official status. The 11 languages can further be broken up into three distinct groups. The two West-Germanic languages, English and Afrikaans (af). Five disjunctive languages, Tshivenda (ve), Xitsonga (ts), Sesotho (st), Sepedi (nso) and Setswana (tn) and four conjunctive languages, isiZulu (zu), isiXhosa (xh), isiNdebele (nr) and Siswati (ss). A key difference between SA disjunctive and conjunctive languages is the former has more words per sentence than the latter. Therefore, disjunctive languages have a higher token count than conjunctive languages. For further details on conjunctive and disjunctive languages with examples, see [1].

The datasets for the ten evaluated languages are available from the South African Centre for Digital Language Resources online repository (https://repo.sadilar.org/). These annotated datasets are part of the NCHLT Text Resource Development Project, developed by the Centre for Text Technology (CTexT, North-West University, South Africa) with contributions by the SA Department of Arts and Culture. The annotated data is tokenized into five phrase types. These five phrase types are:

1. ORG - Organization
2. LOC - Location
3. PER - Person
4. MISC - Miscellaneous
5. OUT - not considered part of any named-entity

The datasets consist of SA government domain corpora. Therefore, the SA government domain corpora are used to do the experiments and comparisons. Eiselen [9] provides further details on the annotated corpora.

III. Language Model Architecture

XLM-Roberta (XLM-R) is a transformer-based multilingual masked language model [2]. This language model trained on 100 languages uses 2.5 TB of CommonCrawl (CC) data [2]. From the 100 languages used by the XLM-R multilingual masked language model, it is noted that Afrikaans (af) and isiXhosa (xh) are included in the pre-training.

The benefit of this model, as indicated by [2] is, training the XLM-R model on cleaned CC data increases the amount of data for low-resource languages. Further, because the XLM-R multilingual model is pre-trained on many languages, low-resource languages improve in performance due to positive transfer [2].

Conneau et al. [2] reports the state-of-the-art XLM-R model performs better than other NN models such as M-BERT and XLM on question-answering, classification, and sequence labelling.

Two transformer models are used for NER evaluation. The XLM-R$_{\text{Base}}$ NN model and the XLM-R$_{\text{Large}}$ NN model. The XLM-R$_{\text{Base}}$ model has 12 layers, 768 hidden states, 12 attention heads, 250 thousand vocabulary size, and 270 million parameters. The XLM-R$_{\text{Large}}$ model has 24 layers, 1024 hidden states, 16 attention heads, 250 thousand vocabulary size, and 550 million parameters [2]. Both pre-trained models are publicly available (https://bit.ly/xlm-rbase, https://bit.ly/xlm-rlarge).

IV. Experimental Settings

The experimental settings for the XLM-R$_{\text{Base}}$ and XLM-R$_{\text{Large}}$ models are described next, followed by the evaluation metrics and the corpora descriptive statistics.

A. XLM-R Settings

The training, validation, and test dataset split was 80%, 10%, and 10%, respectively. Both pre-trained models were fine-tuned with the following experimental settings:

1. Training epochs: 10
2. Maximum sequence length: 128
3. Learning rate: 0.00006
4. Training batch size: 32
5. Gradient accumulation steps: 4
6. Dropout: 0.2

B. Evaluation Metrics

Precision, Recall and F-score are evaluation metrics used for text classification tasks, such as NER. These metrics are used to measure the model's performance during the experiments. The formulas for these metrics leave out the correct classification of true negatives (*tn*) and false negatives (*fn*), referred to as negative examples, with greater importance placed on the correct classification of positive examples such as true positives (*tp*) and false positives (*fp*) [12]. For example, correctly classified spam emails (*tp*) are more important than correctly classified non-spam emails (*tn*). In addition, multi-class classification was used for the research experiments to classify a token into a discrete class from three or more classes. The metric's macro-averages were used for evaluation and comparison. Macro-averaging (*M*) treats classes equally, while micro-averaging (μ) favors bigger classes [12]. Each evaluation metric and its formula as described by [12] are listed below. (*M*) treats classes equally, while micro-averaging (μ) favors bigger classes [12]. Each evaluation metric and its formula as described by [12] are listed below.

Precision$_M$: *"the number of correctly classified positive examples divided by the number of examples labeled by the system as positive"* (1).

$$\frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i + fp_i}}{l} \tag{1}$$

Recall*M*: "*the number of correctly classified positive examples divided by the number of positive examples in the data"* (2).

$$\frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i + fn_i}}{l} \tag{2}$$

Fscore*M*: "*a combination of the above"* (3).

$$\frac{(\beta^2 + 1) Precision_M Recall_M}{\beta^2\ Precision_M + Recall_M} \tag{3}$$

C. Corpora Descriptive Statistics

Table I provides descriptive statistics for the language's training data.

Table I.
The ten languages training data descriptive statistics

Language	Writing System	Tokens	Phrase Types
Afrikaans (af)	Mixed	184 005	22 693
isiNdebele (nr)	Conjunctive	129 577	38 852
isiXhosa (xh)	Conjunctive	96 877	33 951
isiZulu (zu)	Conjunctive	161 497	50 114
Sepedi (nso)	Disjunctive	161 161	17 646
Sesotho (st)	Disjunctive	215 655	18 411
Setswana (tn)	Disjunctive	185 433	17 670
Siswati (ss)	Conjunctive	140 783	42 111
Tshivenda (ve)	Disjunctive	188 399	15 947
Xitsonga (ts)	Disjunctive	214 835	17 904

V. Results and Discussion

A. Results

Table II displays the precision scores of the two XLM-R transformer models compared to models used by [1] and [9]. The Afrikaans (af) language has the highest precision score in this comparison, with 81.74% for the XLM-R$_{\text{Large}}$ model. The XLM-R$_{\text{Base}}$ model has the lowest overall score of 38.59% for the Sesotho (st) language. The CRF model has the highest precision scores for six of the ten languages, including the highest average score of 75.64%. The bold scores in Table II, III and IV show the highest evaluation metric score for each language and the model with the highest average score.

Table III displays the recall scores for the ten low-resourced SA languages. As with the precision evaluation metric, the Afrikaans (af) language has the highest recall score, with 87.07% for the XLM-R$_{\text{Large}}$ model. The XLM-R$_{\text{Base}}$ model has the lowest recall score of 39.41% for the Sesotho (st) language. The CRF and bi-LSTM-aux models have the highest recall scores for three of the ten languages, respectively, with the latter model having the highest average score of 72.48%.

Table IV displays the F-score comparison. The Afrikaans (af) language produced the highest F-score, with an 84.25% for the XLM-R$_{\text{Large}}$ model. The XLM-R$_{\text{Base}}$ model has the lowest F-score of 38.94% for the Sesotho (st) language. The CRF model has the highest F-score for four of the ten languages, including the highest average score of 73.22%.

B. Discussion

The two research questions are answered in this section. The first question is on the transformer model's performance using the three-evaluation metrics, whereas the second question compares the transformer model's performance to the CRF and bi-LSTM models used in the previous SA NER studies.

RQ$_1$ – How does the XLM-R neural network transformer models perform with NER on the low-resourced SA languages using annotated data?

The XLM-R$_{\text{Large}}$ and XLM-R$_{\text{Base}}$ transformer models produced F-scores that ranged from 39% for the Sesotho (st) language to 84% for the Afrikaans (af) language. Further, many of the models recall scores were greater than 70% whereas the precision scores were averaging at 65%. Remember, in this instance, the recall metric emphasizes the average per-named-entity effectiveness of the classifier to identify named-entities, whereas, the precision metric compares the alignment of the classifier's average per-named-entities to the named-entities in the data. All F-scores were above 60% except the Sesotho language, which for both XLM-R models were below 40%. The reason for the low F-scores of the Sesotho (st) language has not been identified, however, it is posited that an investigation into using different hyper-parameter tuning and dataset splits can produce higher F-scores. Sesotho (st) is clearly the outlier during the experiments. For instance, the Sesotho (st) language exclusion from the transformer models results moves the average F-score from 67% to 71%. For the low-resourced SA languages, this is a notable improvement.

RQ$_2$ – How does the XLM-R transformer models compare to other neural network and machine learning models with NER on the low-resourced SA languages using annotated data?

The transformer models were also compared to the findings of previous studies. In particular, [9] used a CRF ML model to do NER sequence tagging on the ten resource-scarce SA languages. Further, [1] implemented bi-LSTM-aux NN models, both with and without embeddings on the same dataset. When analyzing the F-scores, the CRF model has the highest F-scores for four of the ten languages, and the bi-LSTM-aux models shared four of the highest F-scores equally (see Table IV). Meanwhile, the XML-R transformer models have two of the highest F-scores (see Table IV). Although, the transformer models were the only models to produce F-scores greater than 80% for the Afrikaans (af)

TABLE II.
THE PRECISION % COMPARISON BETWEEN TRANSFORMER MODELS AND PREVIOUS SA LANGUAGE NER STUDIES

	Precision				
	CRF*	**bi-LSTM-aux****	**bi-LSTM-aux emb****	**XLM-R**$_{\textbf{Base}}$	**XLM-R**$_{\textbf{Large}}$
af	78.59%	73.61%	73.41%	79.15%	**81.74%**
nr	77.03%	**78.58%**	n/a***	74.06%	73.43%
xh	**78.60%**	69.83%	69.08%	64.94%	65.97%
zu	**73.56%**	72.43%	73.44%	71.10%	71.91%
nso	76.12%	75.91%	72.14%	**77.23%**	n/a****
st	**76.17%**	53.29%	50.31%	38.59%	39.34%
tn	**80.86%**	74.14%	73.45%	67.09%	68.73%
ss	69.03%	**70.02%**	69.93%	65.39%	65.99%
ve	**73.96%**	67.97%	63.82%	58.85%	60.61%
ts	**72.48%**	72.33%	71.03%	63.58%	63.58%
Average	**75.64%**	70.81%	68.51%	65.99%	65.70%

* As reported by [9]. ** As reported by [1]. *** No embeddings were available for isiNdebele.
**** The model was unable to produce scores for Sepedi.

TABLE III.
THE RECALL % COMPARISON BETWEEN TRANSFORMER MODELS AND PREVIOUS SA LANGUAGE NER STUDIES

	Recall				
	CRF*	**bi-LSTM-aux****	**bi-LSTM-aux emb****	**XLM-R**$_{\textbf{Base}}$	**XLM-R**$_{\textbf{Large}}$
af	73.32%	78.23%	78.23%	86.16%	**87.07%**
nr	73.26%	**79.20%**	n/a***	78.51%	78.02%
xh	**75.61%**	73.30%	72.78%	63.53%	64.74%
zu	66.64%	72.64%	74.32%	74.23%	**74.58%**
nso	72.88%	79.66%	77.63%	**80.59%**	n/a****
st	**70.27%**	55.56%	57.73%	39.41%	39.71%
tn	75.47%	**77.42%**	74.71%	73.39%	76.22%
ss	60.17%	71.44%	**72.82%**	70.09%	70.97%
ve	**72.92%**	65.91%	67.09%	63.24%	64.22%
ts	69.46%	**71.44%**	71.25%	68.34%	69.40%
Average	71.00%	**72.48%**	71.84%	69.74%	69.43%

* As reported by [9]. ** As reported by [1]. *** No embeddings were available for isiNdebele.
**** The model was unable to produce scores for Sepedi.

TABLE IV.
THE F-SCORE % COMPARISON BETWEEN TRANSFORMER MODELS AND PREVIOUS SA LANGUAGE NER STUDIES

	F-score				
	CRF*	**bi-LSTM-aux****	**bi-LSTM-aux emb****	**XLM-R**$_{\textbf{Base}}$	**XLM-R**$_{\textbf{Large}}$
af	75.86%	75.85%	75.74%	82.47%	**84.25%**
nr	75.10%	**78.89%**	n/a***	76.17%	75.60%
xh	**77.08%**	71.52%	70.88%	63.58%	64.68%
zu	69.93%	72.54%	**73.87%**	72.54%	73.17%
nso	74.46%	77.74%	74.79%	**78.86%**	n/a****
st	**73.09%**	54.40%	53.77%	38.94%	39.48%
tn	**78.06%**	75.74%	74.07%	69.78%	71.91%
ss	64.29%	70.72%	**71.35%**	67.57%	68.34%
ve	**73.43%**	66.92%	65.41%	60.68%	61.99%
ts	70.93%	**71.88%**	71.14%	65.57%	66.12%
Average	**73.22%**	71.62%	70.11%	67.61%	67.28%

* As reported by [9]. ** As reported by [1]. *** No embeddings were available for isiNdebele.
**** The model was unable to produce scores for Sepedi.

language. This is a significant improvement for NER research on the SA languages.

The comparative analysis identified the Sesotho (st) language as the lowest-performing language across the studies, albeit the CRF model has an F-score of 73%, making it an outlier. If the Sesotho (st) language is excluded from the evaluation, then the metric scores for the transformer models begin to look much different.

For example, the highest average recall score of 72.48% by [1] belonged to the bi-LSTM-aux model, yet, the XLM-R_{Large} model, with Sesotho excluded, was able to produce an average recall score of 73.15%. Similarly, with Sesotho excluded, the average F-score and precision score were 71% and 69%, respectively, which are close to the high scores of the previous studies.

This study reveals that the NN transformer models perform fairly well on low-resource SA languages with NER sequence tagging, and Afrikaans (af) outperforms the other languages using these models. During the NN transformer model experiments, the disjunctive languages had a higher token count, while conjunctive languages had a higher phrase type count (see Table I). However, there is no distinct performance difference between individual disjunctive and conjunctive languages both during the XLM-R experiments and when compared to the other NN and ML models. Nonetheless, except for the CRF model, conjunctive languages had a higher F-score average than disjunctive languages, even with the disjunctive Sesotho (st) language excluded.

The Sesotho (st) language is a clear outlier in this study, with the CRF baseline model F-score being 33% more than the XLM-R models and 18% more than the bi-LSTM-aux models. Interestingly, while both the isiXhosa (xh) and Afrikaans (af) languages were included in the pre-training of the XLM-R model (see Section III) isiXhosa (xh) underperformed when compared to the CRF and bi-LSTM-aux models. This finding suggests including a language in the XLM-R model pre-training does not guarantee good performance during evaluation. It is posited that the experiment results could be improved upon. For instance, additional fine-tuning of the hyper-parameters for each NN model can be done per language, given the available resources. Further, in agreement with [9], the annotation quality could be a contributor to the performance of the models.

VI. Limitations and Further Research

The limitations of this research are the lack of resource capacity to apply additional hyperparameter optimizations on the transformer models per language. Additionally, the named entities of the corpora would need to be investigated and re-evaluated. It is posited, that the quality of the annotations could be improved upon, and the dataset could be re-evaluated using an updated list of named entities.

Additional research, therefore, could implement the transformer models with discrete fine-tuning parameters per language to produce higher F-scores. In addition, the transformer models could be used to evaluate other NLP sequence tagging and sequence-to-sequence tasks such as POS tagging, Phrase chunking, and MT on the low-resource SA languages. Finally, sequence tagging tasks could be evaluated using a linear-complexity recurrent transformer variant.

References

[1] M. Loubser, and M. J. Puttkammer, “Viability of neural networks for core technologies for resource-scarce languages”. *Information*, Switzerland, 2020. https://doi.org/10.3390/info11010041

[2] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, *Unsupervised Cross-lingual Representation Learning at Scale*, 2020. https://doi.org/10.18653/v1/2020.acl-main.747

[3] B. Plank, A. Søgaard, and Y. Goldberg, “Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss”. *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Short Papers*, 2016. https://doi.org/10.18653/v1/p16-2067

[4] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, “Neural architectures for named entity recognition”. *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference*, 2016. https://doi.org/10.18653/v1/n16-1030

[5] J. Lafferty, A. McCallum, and C. N. F. Pereira, “Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data”. *ICML ’01: Proceedings of the Eighteenth International Conference on Machine Learning*, 2001 https://doi.org/10.29122/mipi.v11i1.2792

[6] E. D. Liddy, “Natural Language Processing. In Encyclopedia of Library and Information Science”. In *Encyclopedia of Library and Information Science*, 2001.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need”. *Advances in Neural Information Processing Systems*, 2017.

[8] M. A. Hedderich, D. Adelani, D. Zhu, J. Alabi, U. Markus, and D. Klakow, *Transfer Learning and Distant Supervision for Multilingual Transformer Models: A Study on African Languages*, 2020. https://doi.org/10.18653/v1/2020.emnlp-main.204

[9] R. Eiselen, “Government domain named entity recognition for South African languages”. *Proceedings of the 10th International Conference on Language Resources and Evaluation, LREC 2016*, 2016.

[10] T. Pires, E. Schlinger, and D. Garrette, “How multilingual is multilingual BERT?” *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, 2020. https://doi.org/10.18653/v1/p19-1493

[11] A. Conneau, and G. Lample, “Cross-lingual language model pretraining”. *Advances in Neural Information Processing Systems*, 2019.

[12] M. Sokolova, and G. Lapalme, “A systematic analysis of performance measures for classification tasks”. *Information Processing and Management*, *45*(4), 2009. https://doi.org/10.1016/j.ipm.2009.03.002

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 121–125
DOI: 10.15439/2021F15
ISSN 2300-5963 ACSIS, Vol. 25

In-Bed Person Monitoring Using Thermal Infrared Sensors

Elias Josse, Amanda Nerborg, Kevin Hernandez-Diaz, Fernando Alonso-Fernandez
School of Information Technology (ITE), Halmstad University, Sweden
Email: kevin.hernandez-diaz@hh.se, feralo@hh.se

***Abstract*—The world is expecting an aging population and shortage of healthcare professionals. This poses the problem of providing a safe and dignified life for the elderly. Technological solutions involving cameras can contribute to safety, comfort and efficient emergency responses, but they are invasive of privacy. We use 'Griddy', a prototype with a Panasonic Grid-EYE, a low-resolution infrared thermopile array sensor, which offers more privacy. Mounted over a bed, it can determine if the user is on the bed or not without human interaction. For this purpose, two datasets were captured, one (480 images) under constant conditions, and a second one (200 images) under different variations such as use of a duvet, sleeping with a pet, or increased room temperature. We test three machine learning algorithms: Support Vector Machines (SVM), k-Nearest Neighbors (k-NN) and Neural Network (NN). With 10-fold cross validation, the highest accuracy in the main dataset is for both SVM and k-NN (99%). The results with variable data show a lower reliability under certain circumstances, highlighting the need of extra work to meet the challenge of variations in the environment.**

I. Introduction

The proportion of working-age population is reducing due to increased life expectancy. A large number of people born in the baby-boom of the 60s is now starting to be in need of elderly healthcare. In Sweden, for example, the population beyond 80 years old will increase by 76% in 2035, in parallel to a shortage of healthcare professionals, partly due to a lack of interest in such education amongst young people [1]. As a result, monetary resources (work-related taxes) and dedicated workforce will be proportionally less. There is also a shortage of elderly residences, which is not expected to be solved easily soon [2]. Other countries are facing similar situations too [3].

This raises the issue of taking care of the elderly in both a humane and economically possible way. Nowadays, people get help in their homes. This can be presumed to be even more common, with people wishing to live autonomously as long as possible. Home help can be both human and technological, with current solutions containing a mixture that optimizes personnel resources without sacrificing safety. For example, people can be monitored with sensors at home, specially at night. Staff can get information of whether the person is in bed or not, receiving alerts if e.g. the person leaves the bed too often or for a long time, or simply if just is sleeping poorly.

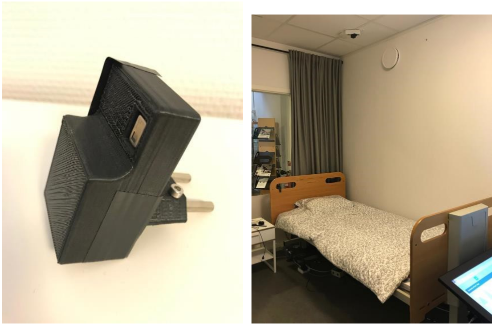


Fig. 1. Left: Griddy with the Panasonic Grid-EYE infrared array sensor on the front. Right: Bed used in data collection with Griddy in the ceiling.

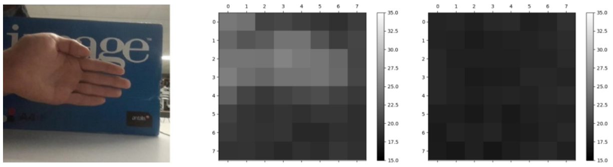


Fig. 2. Left: hand at 30 cm from the camera. Center: capture with Griddy of the scene on the left. Right: capture without hand in the scene.

Cameras in visible range need sufficient light to work. They also pose privacy issues, since people can be recognized. One way is to use low resolution cameras, making recognition difficult. The Grid-EYE sensor from Panasonic offers an 8×8 thermal image that also can operate in darkness [4]. The low amount of pixels offers more privacy. However, it would be impossible for the personnel to monitor by watching, both because of the low resolution, and because it would demand constant attention, which is infeasible if a reduced amount of staff is supposed to monitor a large amount of people.

Accordingly, we are interested in predicting if the person is in bed using low-resolution infrared images, so staff can receive alerts and focus only on those who need help. We use a prototype consisting of a Panasonic Grid-EYE sensor with a Bluetooth module (Fig. 1, left). We call it 'Griddy'. It can be plugged into a 230V socket and collect data at up to 10 frames per second. The data collection environment is a bedroom of our intelligent home [5] (Fig. 1, right). Collection is restricted to a single bed and one person at a time.

A. Related Works

Infrared radiation (IR) is electromagnetic radiation covering a wavelength longer than visible but shorter than millimeters. IR detectors can be categorized into thermal (responsive to

This work has been done by E. Josse, A. Nerborg in the context of their BSc Thesis in Computer Engineering. Authors K. Hernandez-Diaz and F. Alonso-Fernandez thank the Swedish Research Council for funding their research.

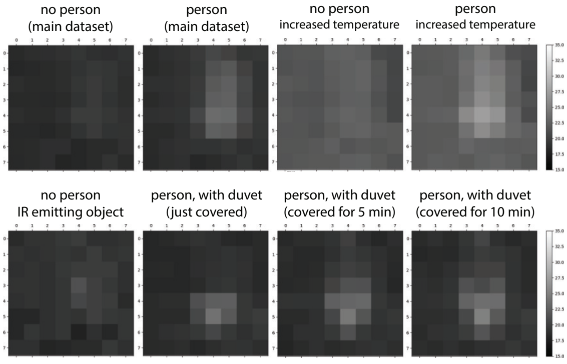


Fig. 3. Mean temperature (per pixel) of several classes.

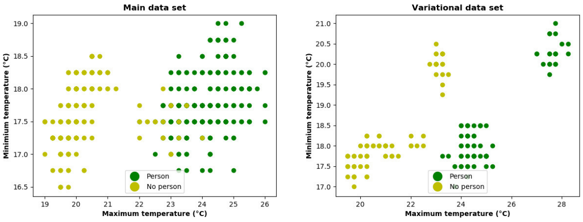


Fig. 4. Scatter plots (maximum and minimum values per image) of the databases for the person/no person images.

heat) and photon (to light) [6], [7]. Passive sensors just capture radiations from objects and are used e.g. to trigger alarms or handle lights automatically. Active sensors transmit and collect the response of radiated elements, allowing to distinguish and track objects [6], [7]. The sensor that we use is of active type.

Low-resolution thermal sensors have been widely used for human detection indoors. Their advantages are low price, privacy preservation, and operation with low light [8], [7]. The Panasonic Grid-EYE has been used in several works. In [8], to improve accuracy by considering temperature variations from other sources than humans. In [7], to detect and track moving humans. In [9], the authors combined Grid-EYE with an ultrasonic sensor (HC-SR04) for fall-detection. The algorithm used, SVM, had the task of differentiating between a fall and another event. In [10], they used the OMRON D6T-44L thermal sensor of 4×4 pixels installed in a bedroom ceiling to recognize body pose and presence, for which they used decision trees. The results indicated that accounting with data with sufficient diversity was of great importance for a good performance. This motivates us to capture data under several environment variations. In [11], they developed a fall detection system using k-NN to classify the body posture. To differentiate between fall and lying down, time differences were used. The body silhouette was used for more privacy.

II. Methodology

A. Software Components

Three machine learning algorithms, Support Vector Machines (SVM) [12], k-Nearest Neighbors (k-NN) [13], and Neural Networks (NN) [14] are used for classification. SVM finds an hyperplane in the feature space that maximizes the margin (minimum distance between the decision boundary and the closest samples, called support vectors). k-NN is one of the simplest machine learning algorithms. It computes the distance of unclassified samples to all samples of the training set, and assigns the class that is most represented in the k nearest neighbours. It is simple, but slow at predicting since it has to compute the distance to the entire training set. NN are modelled after the human brain. They have several layers with a number of neurons per layer. The input is passed through one or more layers (called hidden layers), where the neurons of each layer weight and combine the input of the previous layer, and the output is then passed onto the next layer. The weights of the hidden layers are learned during training, allowing to learn patterns of the input data and model classification functions that can predict the label of a given sample. NN has the ability of handling a large number of training samples, and it is extremely fast in prediction. but it is slow to train.

The code of this work is in Python using the Scikit library. For each algorithm, the most suitable parameters must be found. Four kernels are used for SVM: linear, polynomial, Radial Basis Function (RBF) and sigmoid. A range of different k (k-NN) and number of neurons (NN) are tested to find the optimum ones for our task. The number of neurons in the hidden layers are usually between the size of the input layer (input dimensionality) and the size of the output layer (number of classes). Two hidden layers usually allow to model complex problems with many classes, but there is risk of under-fitting with few data, as in our case [15]. Thus, we will employ one hidden layer, with 1 to 1024 neurons

B. Hardware Components

Grid-EYE, the sensor used, is an active infrared array sensor with 64 thermopile elements arranged in a 8×8 grid. The elements provide a temperature value each. The angle of vision is 60 degrees both horizontal and vertical and the distance of use is up to 7 meters. The output range is 20-100 degrees Celsius, rounded to a quarter degree [4]. Fig. 2 shows an example of capture of a human hand, with a background surface (cardboard) used to protect from inferences.

III. Database and Protocol

Different datasets have been collected, one referred as the main dataset, and the second one referred as the variational dataset. Both have been captured at our intelligent home [reference hidden due to double-blind], as shown in Fig. 1 (right). The bed measures 0.9 × 2 m. Griddy was fixed in the ceiling 2 m over the bed. Thus, the captured area at the bed level is 2.3 × 2.3 m. For the label 'no person', sometimes a person was present just outside the view area. Also, some data was collected from a maximum 30 seconds after a person had been in the bed, while other times it had been several hours. For the label 'person', six different people were presented. They were asked to vary between different sleeping positions to simulate realistic human positioning in the bed. The test group consisted of both male and female adults with different

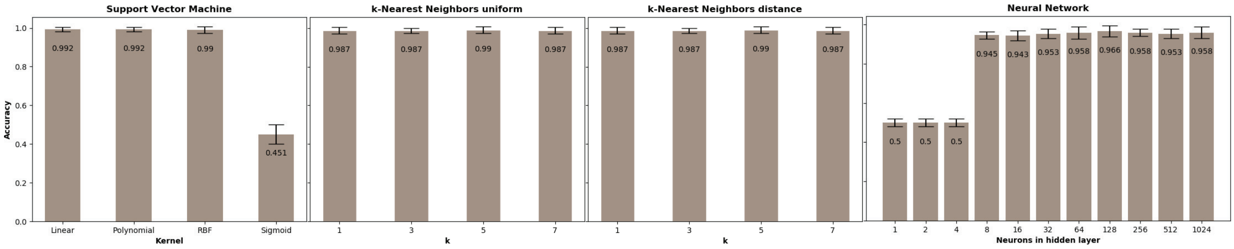


Fig. 5. Accuracy of the different algorithms on the main dataset using 10-fold cross validation. The standard deviation is shown with an error bar.

heights and body types. The aim was to get as much diversity as possible in body temperature, shape, and position.

The *main* dataset has 480 images (240 'person', 240 'no person'). It was collected during 4 different days across 4 weeks. Data of the two classes were equally distributed over the 4 days. Every day, the collection alternated 20 captures of one class and 20 of the other. The order of collection varied from day to day. Fig. 3 (top left) shows the mean temperature of the images of each class (pixel-wise average of all images). Fig. 4 (left) shows the scatter plot of the maximum and minimum values per image of each class. Acquisition across different days and people results in small differences, but the classes are grouped in two clusters, which is expected given the difference in temperature between a person and the room. The data also appears to be linearly separable.

The *variational* dataset incorporates three variations: higher room temperature, a hot non-human object present, and a duvet covering the person. These were chosen because they are expected to occur frequently. For each one, 20 images were collected with a person and 20 without. To increase room temperature, a portable radiator was used to go from the standard 20-21 to 24-25 degrees Celsius. Fig. 3 (top right) shows the mean temperature of the images under this condition. For the second variation, a bottle with warm water at 37 C was used to simulate a small pet. The bottle was placed on top of the bed in various positions. Fig. 3 (bottom left) shows the mean temperature of the images with the water bottle present and no person. The last variation was a duvet. The person was covered up to the neck. Ten images with 'person' were collected right after the person had gotten into bed and covered. Another ten images were collected after five minutes, and another ten after ten minutes. Fig. 3 (bottom right) shows the mean temperature of each class. Finally, the scatter plot of the maximum and minimum values per image of the variational dataset is depicted in Fig. 4 (right). The data forms several clusters given the wider range of variations, specially the maximum temperature, which reaches higher values. Still, the classes appear to be linearly separable. In our examination of the data, the persons appear to heat up the duvet after a while, reaching the same levels than the human body itself. This can be seen in the evolution of the three mean images of Fig. 3 (bottom right). The heat of the water bottle is also detectable when there is no person, but its heat pattern and levels are not equal to those of a person.

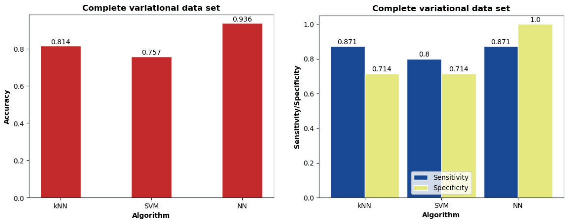


Fig. 6. Accuracy, sensitivity and specificity (variational dataset).

One data sample (input image) consists of a vector of 64 values (number of pixels), which is used as input of the classifiers. The output labels are the classes person/no person. The main data set was divided randomly into 80% (training) and 20% (test), with both classes equally represented. 10-fold cross-validation was used due to the fairly small dataset. The same splits were used for all algorithms and all parameter tweaks, so the same folds were always used. The best configuration of the classifiers (found with the test set) were then retrained on the entire main set, and then evaluated on the variational set.

To compute the success of the predictions, we use the True/False Positive (TP/FP, the system predicts that there is a person in bed, and there is/there is not in reality) and True/False Negative (TN/FN, the system predicts that there is not a person in bed, and there is not/there is in reality). Both FP/FN are system errors, but the consequence of each is different [16]. For example, a FP could lead to a potential critical situation that goes unnoticed (the person has left the bed). A FN, on the other hand, would send an alarm to an operator, when there is no issue in reality. As performance metrics, we employ accuracy, sensitivity and specificity [17]. The accuracy quantifies the right predictions in proportion to the total amount of predictions done, regardless of the actual class, computed as $Accuracy = (TP + TN)/(TP + TN + FP + FN)$. On the other hand, sensitivity describes how well the algorithm predicts positive labels (that the person is in bed), measured as: $Sensitivity = (TP)/(TP + FN)$. Finally, specificity describes how well the algorithm predicts negative labels (that the person is not in bed), as: $Specificity = (TN)/(TN + FP)$.

IV. Experiments and Results

The algorithms are first evaluated on the main dataset to find the best settings (Fig. 5). With SVM, linear, polynomial,

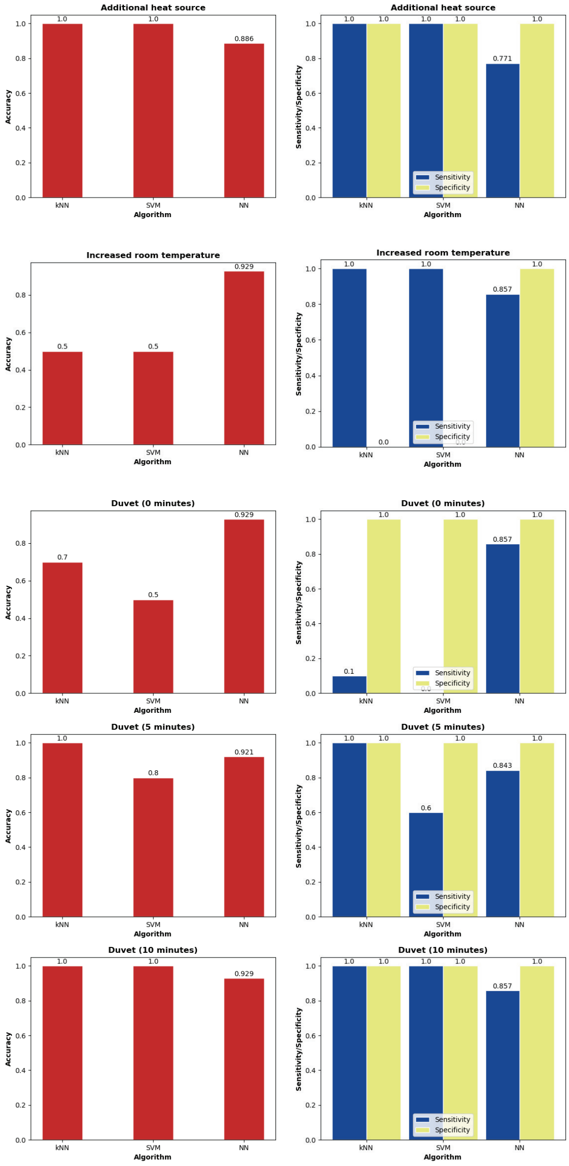


Fig. 7. Accuracy, sensitivity and specificity (subsets of variational dataset).

Radial Basis Function (RBF) and sigmoid kernels are used. For k-NN, we test k=1, 3, 5, 7. We also test two possibilities, one where the k closest neighbours contribute equally to the decision (second column, 'uniform'), and another where the contribution of each neighbour is weighted by the inverse of its distance to the test sample (third column, 'distance'). The uniform approach gives the same importance to each neighbour, while in the distance approach, the closest neighbours are given more importance. With the NN, the number of neurons of the hidden layer is varied from 1 to 1024.

The results show that a high accuracy in general can be obtained with any classifier. The best result with SVM (99% accuracy) is given by several kernels, so for subsequent experiments, we use the linear kernel, since more complex kernels do not show a better accuracy. For k-NN, the accuracy with one neighbour (k=1) is already very high (99%). From these results, the value of k with the variational dataset will be k=1. The experiments between uniform and weighted distances do not show differences either, very likely because the results with k=1 are already nearly to 100% accuracy, so weighting neighbours with their distance does not provide additional gains. Lastly, accuracy with the NN is maximized when 128 neurons are employed (97%), which is the configuration retained for further experiments. Changing the neurons to more or less than 128 has a slight impact, with the accuracy being 94-96%. With less than 8 neurons, the accuracy falls dramatically, being equivalent to tossing a coin (50%).

The algorithms with their best settings are then compared with regards to accuracy, sensitivity and specificity on the variational dataset. Fig. 6 shows the results. The best accuracy is with NN, which is a little below the accuracy on the main dataset (94 vs. 97%). The other two classifiers showed 99% accuracy on the main dataset, but here they go down to 81% (k-NN) and 76% (SVM) With regards to the other metrics, the NN fails in predicting the positive labels (when the person is in bed), with a sensitivity of 87%. When there is no person in bed, it shows a 100% success (specificity). This is good in principle, because the classifier never misses to detect that a person has left the bed. However, on some occasions, there would be false alarms (i.e. the person is really in bed). With the other classifiers, the behaviour is opposite. They have better sensitivity than specificity, which in principle is not as desirable in our scenarios.

We further report the accuracy on subsets of the variational dataset, according to the different variations. Fig. 7 (first row) shows the results with an additional heat source in the form of a filled water bottle, resulting in 100% accuracy with both k-NN and SVM. As seen earlier, these two classifiers were severely affected when passing from the main dataset to the variational dataset, but it seems that this is not the perturbation that produced such change. The NN, on the other hand, sees its accuracy reduced to 89%, so it seems to be more affected by this perturbation. Still the NN has 100% specificity, being the sensitivity the metric that is affected. From this point of view, the capacity of the NN to detect when there is no person in bed goes untouched, but it sees increased its number of false alarms.

The results with increased room temperature from 21-22 to 24-25 degrees Celsius are given in Fig. 7 (second row). Interestingly, k-NN and SVM are severely affected with this perturbation, while in the previous one (temperature increase in only a small region), they performed very well. Its capacity to detect that the person is not in bed (specificity) falls to zero The NN is not as affected as with the previous perturbation, with its accuracy recovered to 93%, and its specificity intact.

Finally, the results with a duvet put at different moments are given in the last three rows of Fig. 7. The performance of the NN (in any of its metric) is independent on the time passed. Its accuracy, sensitivity and specificity is similar to the previous perturbation, and relatively close to the metrics on the entire variational dataset (Fig. 6). This leaves the NN as the best classifier overall, since it appears to be resilient to the majority of perturbations introduced, the only exception being the use of an additional heat source to simulate a pet. With regards to the other two classifiers, we can observe that its accuracy improves as the time with the duvet on increases. With the person just covered (0 minutes ago), they are mostly useless. However, when the person has spent several minutes covered with the duvet, they are capable of obtaining a 100% accuracy (specially k-NN, which achieves that result earlier).

V. Discussion

Technical solutions that contribute to safety, comfort and quick help when needed are essential. The goal of this work is to develop a system that can detect if a bed is occupied or not with an infrared thermal camera placed on the ceiling over the bed. The camera captures images of just 8×8 pixels, which we demonstrate to be sufficient for our purposes, while ensuring that it is not possible to visually distinguish people. This can provide a solution for example to monitor elderly persons on the bed. The person can be monitored with little human interaction, bringing attention of the staff only when there is a potentially dangerous situation, specially at night. This would allow a more effective distribution of resources, for example of car rides to the elderly's home.

We have trained and evaluated three different classifiers, namely Support Vector Machines (SVM), k-Nearest Neighbors (k-NN) and Neural Networks (NN). They have been compared in terms of accuracy (percentage of correct predictions), sensitivity (percentage of correct positive predictions, i.e. person in bed) and specificity (percentage of correct negative predictions, i.e. no person in bed). Overall, the three algorithms behave with a similar high accuracy (97-99%) when trained and tested on the same data conditions. To test the robustness of the system, we have also introduced variations that can be expected in reality, such as a pet sleeping in the bed alongside the person, changes in room temperature, or the person being covered with a duvet after 0, 5 and 10 minutes. The NN shows the best performance overall, being highly resilient to the majority of perturbations, while keeping a specificity of 100%. It means that it is capable of detecting with high accuracy when there is no person in bed under a wide range of perturbations, which is desirable in our scenarios. That the sensitivity is not 100% means that there will be false positives (the person is bed but the system says that is not). However, it is better in principle that the system sometimes wrongfully predicts that the bed is empty (raising alarms that turns out to be false), rather than giving false assurance that the person is in bed. In our case, the sensitivity of the NN is above 84% in the majority of situations. On the other hand, the other two algorithms are highly sensitive to some perturbations, for example increased room temperature, or when a person has just been covered with a duvet. In these cases, their sensitivity or specificity falls to zero (depending on the perturbation). Conversely, with other perturbations, SVM and k-NN show 100% accuracy, which is higher than the corresponding NN accuracy (92-93%).

Such different and opposite behaviour of the classifiers suggests that some sort of classifier combination can be beneficial to cope with image variations, specially with larger databases sizes. As future work, we also plan to expand further the variations of the database. Collection has been constrained to only one person maximum in the image. The person, when present, was always laying on the bed, with no others such as sitting up or standing considered. For some users, it might be the case that a pet is sleeping in the bed alongside the owner. In this case it is of great importance that the system does not wrongfully determine the bed as occupied in the event where the user has left but the pet stayed. A more in-depth analysis is thus needed to determine which kinds of pets and of what sizes the system can handle. Our experiments also show that some classifiers struggle with room temperature variations. We have tested 20-21 and 24-25 degrees Celsius, but improvements need to be done in this regard, including lower and higher temperatures in the experimentation.

It was not possible to visually identify people by looking at the images, or to guess the gender, but it does not mean that is not technologically impossible. It would also not be entirely impossible to determine what activities are taking place in the field of view of the camera. The collected data does not need any manual check, so there is not need to store it over time. Discarding it after interpretation would provide privacy to the user in this regard. The position of the sensor, looking at the bed from the ceiling, may be also controversial. One solution could be to have the sensor on the wall, facing the bed horizontally instead. Another solution could be to cover the room except the bed. If the room is found to be unoccupied, the conclusion would be that the person is on the bed (presuming that the person has not left the home, controlled for example with opportune sensors in the front door).

References

[1] S. S. (SCB), "Stora insatser krävs för att klara 40-talisternas äldreomsorg," https://www.scb.se, accessed 03/2021.

[2] Boverket, National Board of Housing, Building and Planning, "Bostadsmarknadsenkäten," www.boverket.se, accessed 03/2021.

[3] Eurostat, "Ageing europe - statistics on population developments," https://ec.europa.eu/eurostat, accessed 03/2021.

[4] Panasonic, "Infrared array sensor grid-eye," https://industrial.panasonic.com/cdbs/www-data/pdf/ADI8000/ADI8000C66.pdf, accessed 03/2021.

[5] J. Lundström *et al.*, "Halmstad Intell Home," *Proc HealthyIoT*, 2016.

[6] A. Rogalski, *Infrared Detectors*, C. Press, Ed. CRC Press, 2020.

[7] A. D. Shetty *et al.*, "Detection and tracking of a human using the infrared thermopile array sensor — Grid-EYE," in *Proc IEEE ICICICT*, 2017.

[8] A. Trofimova *et al.*, "Indoor human detection based on thermal array sensor data & adaptive backgr. estimation," *J. Comp & Comm* (5), 2017.

[9] Z. Chen, Y. Wang, "Infrared–ultrasonic sensor fusion for SVM–based fall detection," *J. Intell Material Systems and Structures* (29) 2018

[10] B. Pontes *et al.*, "Human-sensing: Low res thermal array classif of location postures," *Dist Ambient & Perv Interactions*, Springer 2017

[11] C.-L. Liu *et al.*, "Fall detect sys w kNN classif," *Expert Sys Appl* 2010

[12] C. Cortes, V. Vapnik, "Support-vector nets," *Mach. Learn.* (20), 1995.

[13] S. Dudani, "Distance-weighted k-nearest-neighb rule," *IEEE TSMC* 1976

[14] E. Alpaydin, *Introduction to Machine Learning*, The MIT Press, 2010.

[15] D. Stathakis, "How many hidden layers & nodes" *J Rem Sensing*, 2009.

[16] A. Burkov, "The Hundred-Page Machine Learning Book" http://themlbook.com/wiki/doku.php, 2019.

[17] D. Altman *et al.*, "Statistics with Confidence Confidence Intervals and Statistical Guidelines," John Wiley & Sons 2013

DOI: 10.15439/2021F28
ISSN 2300-5963 ACSIS, Vol. 25

A Novel Cluster Ensemble based on a Single Clustering Algorithm

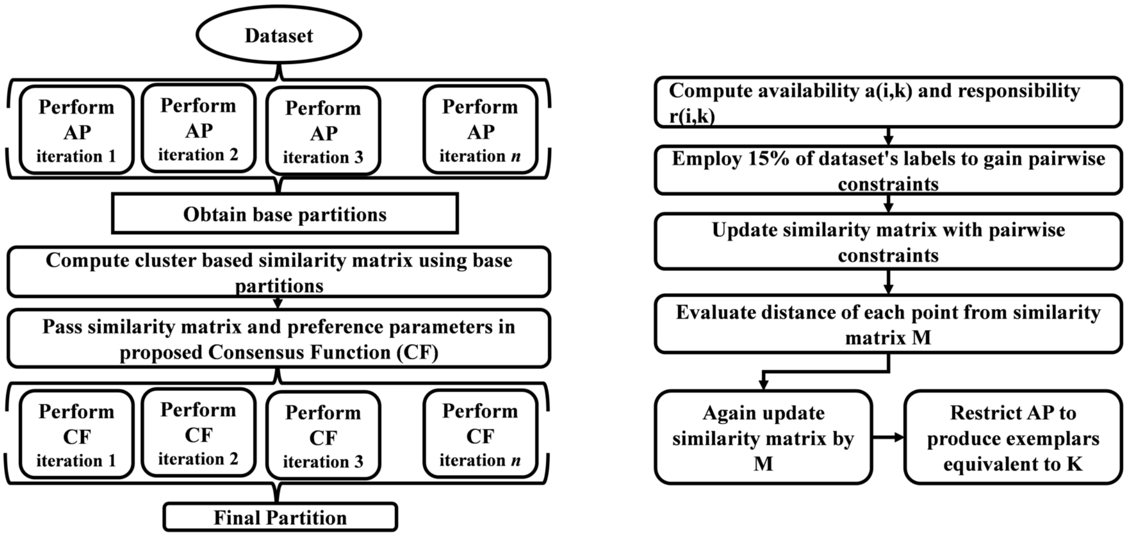
Tahseen Khan[1], Wenhong Tian[1], Mustafa R. Kadhim[1], and Rajkumar Buyya[2,1]
[1]School of Information and Software Engineering, University of Electronic Science and Technology of China, China
tahseen.khan240@gmail.com, tian_wenhong@uestc.edu.cn, mustafa892009@yahoo.com
[2]Cloud Computing and Distributed Systems (CLOUDS) Laboratory
School of Computing and Information Systems, The University of Melbourne, Australia
rbuyya@unimelb.edu.au

***Abstract*—In recent years, several cluster ensemble methods have been developed, but they still have some limitations. They commonly use different clustering algorithms in both stages of the clustering ensemble method, such as the ensemble generation step and the consensus function, resulting in a compatibility issue in terms of working functionality between different clustering algorithms. In addition, in a clustering ensemble method, the accuracy of the final results is a major concern. To deal with it, we propose a novel cluster ensemble method based on a single clustering algorithm (CES). In this method, we iterate a clustering algorithm affinity propagation (AP) ten times in the ensemble generation step to obtain multiple base partitions with a high level of diversity in each iteration due to its nature of producing a random number of clusters. Furthermore, with a few modifications, the same algorithm AP is used to propose a novel consensus function for combining these base partitions into a single partition. The proposed consensus function takes advantage of little side-information in the form of partial labels by using pairwise constraints with AP and number of clusters in a dataset. By employing this information, AP is limited to produce an actual number of cluster centres in a dataset rather than a random number of clusters, which considerably enhanced the accuracy of final outcomes. As a result, CES uses the same clustering functionality in both stages of proposed cluster ensemble method and produces the desired number of clusters in the final partition of a dataset which is significantly improving accuracy when compared to state-of-the-art cluster ensemble methods. Furthermore, as a result of these modifications, the CES outperforms AP in terms of accuracy and execution time. Experiments on real-world datasets from various sources show that CES improves accuracy by 5% on average compared to state-of-the-art cluster ensemble methods and by 55.54% compared to AP while consuming 44.60% less execution time.**

I. Introduction

CLUSTERING is an unsupervised learning technique that seeks to divide a collection of data objects into a set of related classes [1], [2], [3]. It is a crucial and challenging subject in data mining and machine learning, and it has been successfully applied in a wide range of fields, including image processing [4], recommender systems [5], text mining [6], and pattern recognition [7]. A variety of methods have been used in recent years to develop a large number of clustering algorithms [8]. Different algorithms may lead to very different clustering performances for a specific dataset. Each clustering algorithm has its own set of advantages and disadvantages. However, no single algorithm is appropriate for all datasets or applications. Even if a specific algorithm is provided, determining the best parameters for the clustering task can be difficult.

Traditionally, a single clustering algorithm has been used to generate a single clustering result, which has a high rate of inaccuracy. Cluster ensemble has recently emerged as a powerful tool for combining multiple different clustering results (generated by different clustering algorithms or the same algorithm with different iterations) into a potentially better, more robust, and single partition [9]. In detail, a cluster ensemble has mainly two stages: the first, known as the ensemble generation step, obtains multiple base partitions, and the second, known as the consensus function, combines these base partitions [10]. In theory, a functional clustering ensemble must produce reconcilable and well-grounded clustering results when compared to discrete clustering algorithms. However, there were some distinct and demanding issues to deal with while constructing an ensemble for clustering, and it was not as simple as this interpretation suggests. Cluster ensemble is gaining popularity, and several algorithms have been proposed in recent years [11], [12], [13] and [14]. Cluster ensembles can achieve more than a single clustering algorithm in terms of robustness, novelty, stability, and confidence estimation, as well as parallelization and scalability [13]. Despite its considerable success, the current research still faces major challenges. They all have the same flaw: the current cluster ensemble methods use different clustering algorithms in both stages, to obtain base partitions and a final partition, respectively. Furthermore, the use of different clustering algorithms in both stages of the current cluster ensemble architecture may generate compatibility issue related to working functionality. This has motivated us to use a single clustering algorithm in both stages of the new cluster ensemble architecture that significantly improved accuracy of the final outcomes. As a consequence, we propose a novel cluster ensemble method that employs the same clustering in both stages. Accordingly, multiple base partitions are obtained in the its first stage, the ensemble generation process, by executing an unsupervised clustering algorithm affinity propagation (AP) ten times, which provides a high level of diversity among base partitions in each iteration since it generates a random number of clusters [15]. In addition, it also captures all possible different infor-


Dataset
Perform AP iteration 1
Perform AP iteration 2
Perform AP iteration 3
Perform AP iteration *n*
Obtain base partitions
Compute cluster based similarity matrix using base partitions
Pass similarity matrix and preference parameters in proposed Consensus Function (CF)
Perform CF iteration 1
Perform CF iteration 2
Perform CF iteration 3
Perform CF iteration *n*
Final Partition
Compute availability a(i,k) and responsibility r(i,k)
Employ 15% of dataset's labels to gain pairwise constraints
Update similarity matrix with pairwise constraints
Evaluate distance of each point from similarity matrix M
Again update similarity matrix by M
Restrict AP to produce exemplars equivalent to K

1(A): Proposed Cluster Ensemble (CES)

1(B): Proposed Consensus Function (CF)

Fig. 1: 1(A) represents proposed cluster ensemble method, and 1(B) represents proposed consensus function

mation about a data set, which could help improve clustering efficiency. Following that, a similarity matrix is computed between these base partitions, which is known as cluster-based similarity [12]. The computed similarity matrix is then passed as a parameter in the novel consensus function proposed in the second stage of the cluster ensemble method, which uses the same clustering algorithm AP with some modifications. Furthermore, in proposed consensus function, we take advantage of pairwise constraints [16] that employs the concept of must-link (two objects must be in the same cluster) and cannotlink (two objects can not be in the same cluster) with the same clustering algorithm AP to provide a little supervision to the computed similarity matrix. The computed similarity matrix is then updated with this supervised little information, which aids in improving clustering efficiency. At this stage, the similarity matrix is again updated with the Gram matrix, which also enhances clustering efficiency. Furthermore, AP has a flaw in that it generates random number clusters as discussed above. As a result, AP is limited to producing a number of clusters equal to the number of classes in a dataset. This innovative improvement in AP has helped to dramatically increase the accuracy of the final outcomes when this proposed consensus function was used in the proposed cluster ensemble method. As a result, the proposed novel consensus function in cluster ensemble method integrates the base partitions into a single partition. We call our proposed method “A Novel Cluster Ensemble based on a Single Clustering Algorithm (CES)”, because we use the same functionality in each stage of it, as shown in Figure 1(A). CES’s key benefit is that it eliminates the complication of using two separate clustering paradigms in both stages, making it compatible, and improving clustering outcomes such as accuracy over stare-of-the-art cluster ensemble methods. In addition, when compared to AP, the innovative change significantly improves accuracy and execution time.

This paper makes the following key contributions:

- We propose a novel cluster ensemble method based on a single clustering algorithm , while conventional cluster ensemble methods use different clustering algorithms in both stages, resulting in compatibility issue in ensemble generation and consensus function.
- We propose a novel consensus function based on AP that integrates pair-wise constraints, Gram matrix, and limits AP to produce the actual number of clusters present in the dataset.
- The proposed cluster ensemble method outperforms AP in terms of accuracy and execution time.

The rest of the paper is organized as follows: Section II formulates the background of our work and defines consensus clustering problem. Section III provides details of the proposed framework with selected clustering algorithm AP. Section IV presents the experiments carried out for the framework on different real-world data sets and comparatively explains results. Finally, Section V concludes the paper and reveals the limitation of our work and ongoing work to overcome it.

II. Related Work

A clustering ensemble combines multiple base partitions obtained in ensemble generation step into a robust, accurate and single partition by using a consensus function [11]. The advantage of using cluster ensemble is that it increases the accuracy of the outcomes by taking individual solution biases into account. [17] was the first to propose three cluster ensembles. The first was the cluster-based similarity partitioning algorithm (CSPA), which was based on data point similarity *S*, with *S* modified according to whether data points are similar or dissimilar. The hypergraph partitioning algorithm (HGPA) was the second, which was based on re-partitioning data using the given clusters. The final one was the meta-clustering algorithm (MCLA), which was based on clustering clusters and rendered each cluster by a hyperedge. [12] proposed the Adaptive Clustering Ensemble (ACE), which consisted of three stages: the first was to convert the base clusters into binary representations. The second stage was to find similar clusters based on cluster-based similarity, and the third was to obtain consensus function results by dealing with uncertain objects in order to achieve better final consensus clustering partitions of data. Furthermore, many proposed cluster ensembles has been proposed recently, for example, quadr mutual information consensus function (QMI), mixture model (EM) [13]. QMI is a consensus function based on quadratic mutual information, which is proposed and reduced to k-means clustering in the space of specially altered cluster labels. EM is unsupervised decision-making fusion method based on a probability model of the consensus partition in the space of contributing clusters. [11] proposed the weighted spectral cluster ensemble (WSCE) as a new cluster ensemble focused on group detection arena and graph based clustering concepts. Multiple base partitions are obtained using a new version of spectral clustering and combined into a single robust partition using a proposed consensus function in this method. [14] proposed a cluster ensemble method based on distribution cluster structure, with final results produced using a newly proposed distribution-based normalised hypergraph cut technique. [18] proposed two new cluster ensemble methods: ensemble clustering by propagating cluster-wise similarities with hierarchical consensus function (ECPCS HC) and ensemble clustering by propagating cluster-wise similarities with meta-cluster based consensus function (ECPCS MC). Some research has centred on the applications of cluster ensembles in different areas, for example, time series analysis has become a popular research topic in the field of pattern recognition, particularly for detecting manufacturing flaws. As a result, [19] proposed an automated alternative called control chart pattern recognition (CCPR) model based on consensus clustering. Furthermore, [20] proposed a cluster ensemble method for unsupervised pattern recognition that centred on the growth of damages in composites under solicitations.

The following notations will be used consistently in this paper. Table I also contains several important notations with their definitions that were used in this article. We call a set of objects $D = \{x_1, x_2, ..., x_n\}$, where each object $x_i \in D$ is represented by a vector of N attribute values $x_i = (x_{i,1}, ..., x_{i,N})$. Let $\Gamma = \{\beta_1, \beta_2, ..., \beta_m\}$ be a cluster ensemble with m base partitions, where each base partition is an "ensemble member", and returns a set of clusters $\beta_h = \{\beta_1^h, \beta_2^h, ...\beta_n^h\}$, such that $\bigcup_{p=1}^{k_h} \beta_p^h = D$,where k_h is the number of h^{th} clustering. For each data point $x_i \in D, \beta^h(x_i)$ indicates cluster label in the g^{th} base partition to which data point x_i belongs to, i.e. $\beta^h(x_i) = \beta_h^p$, if $x_i \in \beta_h^p$. As a result, the problem is to find a new partition $\Gamma^* = \beta_1^*, \beta_2^*, ...\beta_K^*$, where K is the number of clusters in the final clustering result of the dataset D, which summarises the details from the cluster ensemble Γ [21].

TABLE I: Important notations used in this paper

Definition	**Symbol/Notation**
Dataset	D
Data object	$x_i \in D$, $1 \leq i \leq n$
Number of objects	n
Number of ensemble members	m
Ensemble member	β_i, $1 \leq j \leq m$
Similarities between objects	S_{ij}, $1 \leq i \leq n$, $1 \leq j \leq n$
Distance from similarity matrix	P_{ij}, $1 \leq i \leq n$, $1 \leq j \leq n$
euclidean distance	d_{euc}
Similarities between ensemble members	S_m
Preference parameter for ensemble members	p_m

III. A Novel Cluster Ensemble based on a Single Clustering Algorithm

Figure 1A depicts the proposed cluster ensemble method which consists of two steps: (1) an ensemble generation step in which multiple base partitions are obtained by running AP ten times; (2) a proposed consensus function using AP that combines these multiple partitions into a single robust partition. The proposed cluster ensemble method's operation is described in more detail below. Algorithm 1 presents the pseudo code of CES.

A. First Stage: Ensemble Generation Step

The first step is called ensemble generation, and our main goal is to generate m base clustering members. In algorithm 1, steps from 2 to 5 represent the ensemble generation step. Any clustering algorithm can be used to generate ensemble members as long as it produces as many different members as possible [12]. At this stage, different partitions of the same dataset can be created using independent runs of different clustering algorithms or the same clustering algorithm [22][9][18]. Then, in the following stage, a consensus function is used to obtain a final partition from the base partitions generated in the previous stage. Accordingly, we use unsupervised AP, as described in Section III-B1, and run it $(iter = 10)$ times to create multiple m ensemble members, such that $\beta_i \in \Gamma$, where $i \in (1, ..., n)$ and n are the number of data objects. The reason for AP's adoption is that it generates a random set of exemplars (clusters) in β_h, where β_h is an ensemble member, which provides a high level of diversity among ensemble members in each iteration and acquires all possible distinct information about a data set, which may help to increase clustering performance. In other words, in each iteration, AP offers distinct clusters, ensuring the foundation of ensemble

Algorithm 1: The pseudo code of our proposed cluster ensemble method CES

```
Input: data, No. of clusters $K$
Output: the clustering Outcomes $\Gamma^*$
1: $no\_classes \leftarrow K$, $random \leftarrow []$, $temp \leftarrow []$, $O \leftarrow []$,
   $s \leftarrow []$ $Z \leftarrow []$, $idx \leftarrow []$, $status \leftarrow []$, $availability \leftarrow a_{ik}$,
   $responsibility \leftarrow r_{ik}$
2: Calculate $m$ base partitions $\beta_i$ by executing AP ten times
3: $S_m \leftarrow Euclidean(\beta_i, \beta_i)$ /* where $S_m$ is
   similarity matrix */
4: $p_m \leftarrow min(S_m)$ /* where $p_m$ is preference
   parameter */
5: Pass $S_m$ and $p_m$ in proposed consensus function
   /* Proposed Consensus Function
   (modified AP)) */
       /* Execute consensus function ten
   times */
6: Compute $a_{ik}$ and $r_{ik}$
7: $s \leftarrow .15(labels)$
8: for $i$ = 1 to length(s) do
       for $j$ = $i$ + 1 to length(s) do
           if $(x_i, x_j) \in C$ then
               $status \leftarrow 0$
           else
               $status \leftarrow 1$
           /* where $C$ denotes
           cannot-link constraints */
9: return $status$
10: $S_{ij}$ & $S_{ji}$ = $status$ /* where $i \in (1,...,n)$,
    $j \in (1,...,n)$ */
11: $P_{ij} \leftarrow \frac{S_{1j}^2+S_{i1}^2+S_{ij}^2}{2}$ /* where $i \in (1,...,n)$, $j \in (1,...,n)$
    */
12: $S_{ij} \leftarrow P_{ij}$ /* where $i \in (1,...,n)$, $j \in (1,...,n)$ */
13: $Z \leftarrow$ set of exemplars
14: $Z \leftarrow Sort(Z, descending)$
15: if $length(Z) < no\_classes$ then
        $no\_classes \leftarrow length(Z)$
16: $random \leftarrow Random(length(Z), no\_classes)$
17: $O \leftarrow Z[random]$
18: for $i$ = 1 to $no\_classes$ do
        for $j$ =1 to length(Z) do
            $temp \leftarrow Z[j]$
        if $temp$ = $O(i)$ then
            $idx \leftarrow temp$
19: return $idx$
20: $\Gamma^* \leftarrow idx$
```

clustering, which is that ensemble members should have a high level of diversity to capture all of a dataset's information. [12].

Definition 1: Let $X = (X_1, X_2, ...X_N)$ and $Y = (Y_1, Y_2, ...Y_N)$ are two points in euclidean N-space, then Euclidean Distance d_{euc} from point X to Y and Y to X is given by Equation (1) from [23]:

$$\begin{aligned} d_{euc}(X,Y) &= d_{euc}(Y,X) \\ &= \sqrt{(Y_1-X_1)^2+(Y_2-X_2)^2+...+(Y_N-X_N)^2} \\ &= \sqrt{\sum_{i=1}^{N}(X_i-Y_i)^2} \end{aligned} \qquad (1)$$

where X and Y represent two vectors in euclidean N-space that begin at the space's origin.

Thus, the lower the d_{euc} value between two sets of observations, the more similar they are and the more likely they are in the same cluster. As a result, we use this method to combine the m base partitions found in Section III-A. We use the Euclidean distance, as discussed above in Equation (1), to compute similarities between pairs of ensemble members. The similarities between ensemble members is known as cluster-based similarity. So, as shown in Equation (2), the S_m similarities for m ensemble members can be computed:

$$S_m = \sqrt{\sum_{i=1}^{m}(\beta_i-\beta_j)^2} \qquad (2)$$

for all $i \in \{1,...,m\}$ and $j \in \{1,...,m\}$. As a consequence, the base partitions are derived as similarities between m ensemble members, and these base partitions are then grouped into a single partition using the proposed consensus function in Section III-B2. For this, we pass S_m and $p_m = min(S_m)$ in the proposed consensus function parameter, which is proposed using AP.

B. Second Stage: Consensus Function

The consensus function, which is responsible for achieving the final partition of the data by using base partitions generated during the ensemble generation step, is another important component of the cluster ensemble method. We propose a very effective and efficient consensus function, as explained in the sections below, because the consensus function has a direct impact on the performance of the cluster ensemble method. In algorithm 1, steps from 6 to 20 represent the consensus function step. The main idea behind proposing a new consensus function is to compute cluster-based similarities between pairs of ensemble members or clusters rather than computing similarities between data objects [12]. The proposed consensus function's operation is discussed further below. In Section III-B1, we describe some information about the traditional clustering algorithm AP, and then in Section III-B2, we show how it is improved and used in proposing the consensus function.

1) Affinity Propagation (AP): Affinity Propagation (AP)[15] is a clustering algorithm that works on the principle of message passing between data objects. Unlike other clustering algorithms such as k-medoids or k-means, AP does not seek to determine the number of clusters before running the algorithm. AP, like k-medoids, seeks "exemplars," or members of the input set that are representative of clusters. In other words, rather than taking the number of clusters K as input, AP takes the collection of real-valued similarities S_{ik}, which indicate how well data object at index k is suited to be an exemplar for data object i for two data objects $(x_i,x_k) \in D$. In addition, AP accepts real numbers S_{kk} as input, with the possibility of selecting high similarity data objects as exemplars (number of clusters), referred to as preference p. The exemplars are influenced not only by p but also by message passing. This value can be changed to generate a different number of clusters. Moreover, this value can be a median of the input collection of real-valued similarities that yields a moderate number of clusters or a minimum of these that yields the fewest clusters. Additionally, two real-valued messages which are the 'responsibility' r_{ik} from data object x_i to x_k that depicts how well deserved the data object x_k is to serve as the exemplar of data object xi and the 'availability' a_{ik} from data object x_k to x_i that depicts how suitable it would be for data object x_i to select x_k as its exemplar, are computed. r_{ik} and a_{ik} can be considered as log-probability ratios. Initially, availabilities a_{ik} were set to zero: $a_{ik} = 0$. The responsibilities r_{ik} are then computed using Equation (3).

$$r_{ik} \leftarrow S_{ik} - \max_{k'\, s.t.\, k' \neq k}\{a_{ik'} + S_{ik'}\} \tag{3}$$

Because a_{ik} is set to 0 in the first iteration, r_{ik} has been assigned the difference of s_{ik} and the largest of the similarities between the data object at index i and the other candidates. As a result, if some data objects are assigned to exemplars in subsequent iterations, their availabilities a_{ik} fall below zero, as shown by the Equation (4). And these negative availabilities will have an effect on the similarities $S_{ik'}$ in Equation (3), and the corresponding exemplar will be removed from the competition. And in the Equation (3), for $i = k$, the responsibilities become r_{kk}, which is equivalent to input preference and point at indexed k or i is chosen as an exemplar. This condition allows other candidate exemplars to compete to be an exemplar for a data object and updates availabilities using Equation (4) below.

$$a_{ik} \leftarrow \min\left\{0, r_{kk} + \sum_{i'\, s.t.\, i' \notin \{i,k\}} \max\left\{0, r_{i'k}\right\}\right\} \tag{4}$$

Thus, in Equation (4), availabilities a_{ik} are assigned to the sum of self-responsibility r_{kk} and positive responsibilities received by the candidate exemplar at index k from other data objects. Only positive responsibilities are added here because it is required for a good exemplar. If self responsibility becomes negative, the availability of data objects at index k can be increased, and self-availability a_{kk} is updated using Equation (5).

$$a_{kk} \leftarrow \sum_{i'\, s.t.\, i' \notin k} \max\{0, r_{i'k}\} \tag{5}$$

As a result, these messages are exchanged between two data objects with pre-computed similarities. At any point, availabilities and responsibilities can be combined to identify a potential exemplar. As a result, $(a_{ik} + r_{ik})$ should be the maximum to determine which data object at index i should be chosen as an exemplar. And knowing $i = k$ leads to knowing the data object that is an exemplar for the data object at index i.

2) Proposed Consensus Function: In proposed consensus function, we take advantage of little side-information such as pairwise constraints [16], which are made up of two constraints: must-link and cannot-link. It has helped to increase the precision in accuracy. We assume that partial class information is provided in the form of pairwise constraints showing whether two objects are members of the same ($must - link$ constraint) or different ($cannot - link$ constraint) clusters. The cluster information is expressed via a set $\Psi \subset D \times D$ $m_l = \{x_i, x_j\}$ where $\Psi = M \cup C$, and

$$\begin{aligned} M &= \{(x_i,x_j) \in D \times D : x_i \text{ and } x_j \in \text{same cluster}\} \\ C &= \{(x_i,x_j) \in D \times D : x_i \text{ and } x_j \in \text{different clusters}\} \\ &\qquad \text{where } i, j \in (1,2,...,n) \end{aligned} \tag{6}$$

Let us say we have pairwise constraints for some data objects and want to incorporate this side-information into our model. The first question is where we can use this side-information. One approach could be to directly connect the hidden variables corresponding to data points that must be in the same cluster via a function that applies the constraints, and to connect the hidden variables corresponding to cannot-link data objects via a suitable function [24]. Another approach could be to manipulate the similarities between the data objects. If two data objects are in the same cluster, we can maximise their similarities and minimise them if they are in different clusters. As a result, we can conclude that clustering performance is directly related to the similarities between data objects.

Definition 2: Let us suppose there two data objects such that $(x_i,x_j) \in D$ where $i \in (1,2,...,n)$, $j \in (1,2,...,n)$, the similarities between these objects S_{ij} or S_{ji} will be adjusted according to Equation (7) below.

$$\begin{aligned} (x_i,x_j) \in M &\Rightarrow S_{ij} = 1 \,\&\, S_{ji} = 1 \\ \text{and } (x_i,x_j) \in C &\Rightarrow S_{ij} = 0 \,\&\, S_{ji} = 0 \end{aligned} \tag{7}$$

As a result, this adjustment in similarities can increase more supervision to improve clustering performance because it increases the probability of similar constraints being in the same cluster as much as possible. As discussed in section III-B1, AP takes as input a collection of similarities between data objects and a preference that can be the median or minimum of the input similarities; unlike other algorithms such as k-means and k-medoids, it does not take the number of exemplars K as input. In addition, after exchanging real-valued messages, it generates a random number of exemplars to compute a_{ik} and r_{ik}, which may affect its clustering performance. So, to solve this problem, we use the number of exemplars K as an input parameter in AP. After that, real-valued messages a_{ik} and r_{ik}

are computed. At this point, we include the concept of pairwise constraints, and 15% of the actual labels were enforced to know constraints for each pair of data objects, and similarities are updated as a result. From Section III-A, we already have S_m and p_m in the AP's parameter. Therfore, S_m is iteratively updated with 1 (if they are in the same cluster) or 0 (if they are not) (if they are in different clusters), for two data objects $(x_i, x_j) \in D$, where $i \in (1,2,...,n)$ and $j \in (1,2,...,n)$.

Definition 3: Let S_{ij} comes from distances between data objects, then there are $x_i \in R^m$, then a matrix P_{ij} from distance matrix S_{ij} can be defined as:

$$P_{ij} = \frac{S_{1j}^2 + S_{i1}^2 + S_{ij}^2}{2} \tag{8}$$

for $i \in \{1,\ldots,n\}$ and $j \in \{1,\ldots,n\}$. P_{ij} is a positive semi-definite matrix of rank at most two which is known as Gram Matrix.

After adjusting similarities with constraints, new similarities are again updated with Gram Matrix as shown in Equation (9).

$$S_m \Leftarrow P_{ij} \tag{9}$$

This motive has come with enhancement in clustering accuracy when this consensus function has been utilized in our proposed cluster ensemble method CES. Finally, a good set of exemplars is obtained by using the updated similarities, as shown in Equation (9). At this point, we solve the previously discussed unsupervised AP problem, which generates a random number of exemplars. We use side-information such as the number of exemplars K passed as input to AP and restrict it to generate exemplars equivalent to K by iterating the obtained fine set of exemplars. As a result, AP clustering accuracy and execution time are dramatically improved. Thus, as shown in Figure 1B, we present a novel consensus function that is used in our cluster ensemble method CES, as shown in Figure 1A. Finally, a single robust dataset partition is produced in Γ^* equivalent to the number of clusters in the dataset.

TABLE II: Real-world data sets taken from different sources

S.No	Dataset	number of objects	Features	Classes
1.	aerosol	905	892	3
2.	alphabet	814	892	3
3.	aquarium	922	892	3
4.	banana	840	892	3
5.	basket	892	892	3
6.	blog	943	892	3
7.	book	896	892	3
8.	heartdisseaseh	294	13	5
9.	glass	214	10	6
10.	heap	155	19	2
11.	wing	856	899	3
12.	water	922	899	3

IV. Performance Evaluation

A. Experimental Design

The proposed clustering ensemble method CES is compared to several representative clustering ensemble methods on a variety of real-world data sets using representative assessment criteria to assess its performance. Our method is tested in ten separate runs. We choose a standard evaluation criterion, such as micro-precision, to assess its performance, which compares real labels to predicted labels to assess clustering approaches' accuracy [29]. [25] has evaluated the consensus cluster's accuracy in terms of true labels using micro-precision. This assessment criteria is also taken into account by [26]. As a result, we have used the only considered evaluation criterion to compare the CES approach to other clustering approaches in order to further evaluate its performance. The following are the remaining paragraphs in this section: The datasets used for comparisons will be addressed first. Then we will go over the assessment criteria and the steps of the experiment in detail.

We choose a variety of real-world data sets to implement the experimental study of the proposed CES approach, which are described in Table II. The twelve real-world data sets, which include different samples, features, and classes, were gathered from various sources, including the UCI repository and the Microsoft Research Asia Multimedia (MSRA-MM) image dataset obtained from Microsoft [30]. These data sets are also used in classification due to the availability of class labels, but class labels are not used in clustering for the evolutionary process of clustering [31]. We use micro-precision to assess the accuracy of the consensus cluster with respect to the true labels. If a data set has K classes and n objects, the micro-precision m_p is defined as in Equation (10):

$$m_p = \sum_{i=1}^{K} \left[\frac{a_i}{n} \right] \tag{10}$$

where a_i represents the number of items in consensus cluster i, and $0 \leq m_p \leq 1$ represents the best possible consensus clustering that is analogous to class labels. As a result, we can assume that the higher the m_p value, the better the clustering performance.

Matlab R2019a was used to design the experiment. Our experiment is divided into two phases: generating ensemble members for these real-world datasets using the clustering algorithm AP, and obtaining consensus function results using the proposed consensus function described in Section III-B2. To begin, a similarity matrix is computed using pairwise euclidean distance and the number of objects n and features f in a dataset, yielding a $n \times n$ similarity matrix S. The preference parameter p is then set to $p = \min(S) \,/\, iter \times 0.3$, where $iter$ denotes the iteration number for this step, which is set to 10 to produce m ensemble members. The value $iter \times 0.3$ is used to generate various base partitions and has an impact on clustering performance. The similarity matrix Sm is computed using these acquired base partitions and the preference parameter is set to $p_m = \min(S_m))\,/\, iter \times .09$ after receiving m base partitions after 10 execution of unsupervised

TABLE III: Comparison of Accuracy evaluated using micro-precision between CES and other cluster ensemble methods

Dataset	**CES**	CSPA	HGPA	MCLA	WSCE	EM	QMI	ECPCS MC	ECPCS HC
aerosol	**54.03**	50.28	50.28	50.28	51.27	39.67	50.61	53.26	51.05
alphabet	**51.97**	47.30	47.30	47.30	47.30	37.59	48.40	47.91	48.16
aquarium	**70.17**	**70.17**	**70.17**	**70.17**	69.63	36.23	70.07	65.73	69.96
banana	**47.98**	42.74	42.74	42.74	44.29	39.40	44.17	43.57	43.21
basket	**56.28**	56.05	56.05	56.05	**56.28**	37.89	55.83	52.58	**56.28**
blog	**73.59**	73.49	73.49	73.49	72.64	35.42	73.49	66.49	73.49
book	**57.70**	57.48	57.48	57.48	57.59	36.27	57.48	56.70	57.37
heartdisseaseh	**66.33**	63.95	63.95	63.95	50.00	30.27	54.08	55.10	57.82
glass	**65.42**	35.51	35.51	35.51	58.88	45.79	45.79	52.34	52.80
heap	**79.35**	54.84	54.84	54.84	77.42	59.35	59.35	59.35	58.71
wing	**62.03**	61.92	61.92	61.92	61.68	37.38	6168	57.59	61.68
water	**57.16**	56.94	56.94	56.94	56.29	36.66	56.62	55.86	57.05
Avg	**61.83**	55.89	55.89	55.89	58.61	39.33	56.46	55.54	57.30

TABLE IV: Accuracy and Execution time (seconds) between CES and AP

(a) Comparison of Accuracy between CES and AP

Dataset	AP	CES
aerosol	20.99	54.03
alphabet	15.36	51.97
aquarium	15.08	70.17
banana	18.21	47.98
basket	27.35	56.28
blog	19.72	73.59
book	22.99	57.70
heartdisseaseh	39.80	66.33
glass	53.74	65.42
heap	59.35	79.35
wing	22.90	62.03
water	14.43	57.16
Avg	27.49	61.83

(b) Comparision of Execution time between CES and AP

Datasets	AP(Avg)	AP(Max)	CES(Avg)	CES(Max)
aerosol	5.6822	6.2422	2.6765	2.7853
alphabet	1.9892	2.8138	1.8246	1.8878
aquarium	2.2208	3.1873	2.2521	2.3075
banana	3.5394	4.6083	1.9807	2.0751
basket	5.0143	5.5720	1.9964	2.0163
blog	1.9467	2.9702	2.2170	2.2603
book	1.8947	2.2981	2.1448	2.2040
heartdisseaseh	0.3843	0.6017	0.4923	0.5328
glass	0.3037	0.5951	0.2521	0.2709
heap	0.1677	0.4678	0.1893	0.2066
wing	1.9915	2.8967	1.9963	2.0228
water	4.7009	5.4571	2.2588	2.3218
Avg	2.4863	3.1425	1.6901	1.7409

TABLE V: Comparison of Accuracy between CES and other work with common datasets and evaluation criteria micro-precision

Study	Dataset	Accuracy
CES	blog	73.59
[25]		71.14
CES	aquarium	70.17
[25]		68.56
CES	glass	65.42
[26]		61.21
CES	glass	65.42
[27]		64.40
CES	glass	65.42
[28]		47.20

AP. These parameters, as well as the number of classes K, are passed as input parameters into the proposed consensus function for further calculations to determine final partitions of a dataset in K clusters. The introduced consensus function is also executed with $iter = 10$. The primary goal of this experiment is to evaluate the performance of CES and to see how effective our algorithm is when compared to other traditional clustering ensemble methods such as (CSPA, HGPA, MCLA [17]), (EM, QMI [13], WSCE [11], (ECPCS MC, ECPCS HC [18] by micro-precision. CES also outperforms AP in terms of accuracy and execution time due to innoative changes.

B. *Results and Discussions*

The accuracy of CES and other traditional cluster ensemble techniques are tested on real-world data sets derived from different sources measured by micro-precision is shown in Table III. Table IV shows the accuracy and execution time evaluated between AP and CES. The experimental results are explained in two parts: (1) comparisons on real-world data sets for accuracy between CES and other cluster ensemble methods, and (2) comparison of accuracy and execution time between AP and CES.

As a result, it is concluded that, when compared to other clustering ensemble methods, CES has achieved promising results in accuracy assessment on all datasets, as shown in Table III. Although CSPA, HGPA, MCLA, and CES achieved comparable accuracy of 70.17% in the dataset aquarium, WSCE, ECPCSHC, and CES also achieved comparable accuracy of 56.28% in the dataset basket, CES outperformed state-of-the-art clustering ensemble methods WSCE, ECPCSMC and ECPCSHC by 5.21%, 6.29% and 4.53% on average respectively. Furthermore, CES has also outperformed all cluster ensemble methods by 5% on average. The use of the same clustering functionality in both cluster ensemble steps may boost the stability of clustering results, resulting in a significant improvement in clustering accuracy. We see a significant improvement in high-dimensional data sets with noises, such as aerosol, alphabet, aquarium, banana, basket, blog, book, wing, and water, because we limit AP to produce the actual number of clusters in the proposed consensus function. Furthermore, the clustering accuracy has been compared to state-of-the-art cluster ensemble methods that use common data sets and evaluation criterion micro-precision shown in Table V. The clustering ensemble approach HCEKG by [25] has achieved approximately 71.14% and 68.56% clustering accuracy with the blog and aquarium datasets, respectively, whereas our CES has obtained 73.59% and 70.17% indicating 3.33% and 2.29% improvement respectively. [26] has achieved 61.21% accuracy with glass dataset while CES has achieved 65.42%, indicating a 6.45% improvement. With the glass dataset, [27] has achieved 64.4% accuracy, while CES has achieved 65.42%, indicating a 1.56% improvement. [28] has obtained 47.20% accuracy with basket dataset, while CES has obtained 65.42% with a 27.85% improvement.

CES has significantly improved in terms of accuracy and execution time when compared to AP. Table IVa clearly shows that CES achieved a significant improvement in clustering accuracy and execution time when compared to AP. Furthermore, CES has achieved an average accuracy of 61.83% across all twelve datasets, whereas AP has achieved an average accuracy of 27.49% with a 55.54% improvement. When it comes to execution time, CES has significantly outperformed AP as shown in Table IVb. We have measured execution time on various real-world datasets with low and high dimensions, including (heartdisseaseh, 13), (glass, 10), (heap, 19), and (aerosol, 892), (alphabet, 892), (aquarium, 892), (banana, 892), (basket, 892), (blog, 892), (book, 892), (wing, 899) and (water, 899). When considering the maximum time in 10 iterations, CES has consumed 3.4569 seconds, 0.926 seconds, 0.8798 seconds, 2.5332 seconds, 3.5332, 3.5557 seconds, 0.79099 seconds, 0.0941 seconds, 0.0689 seconds, 0.3242 seconds, 0.2612 seconds, 0.8739 seconds, and 3.1353 seconds less than AP. Finally, CES took 1.4016 seconds less than AP on all real-world datasets; additionally, our method has consumed 44.60% less execution time than AP. When it comes to average time, AP outperforms on some of the datasets, but only by a small margin. Nonetheless, when the average performance of average time consumed on all datasets is considered, CES has consumed 32.02% less time than AP. The proposed cluster ensemble method, depicted in Figure 1(A), has quadratic time complexity, i.e., in $O(n^2)$ time, whereas the proposed consensus function, depicted in 1(B), has time complexity of order $O(n^2)$ i.e., $O(n^2+n)$ time.

V. Conclusions and Future Work

In this paper, we proposed a new cluster ensemble method (CES), which is capable of dealing with limitations of traditional cluster ensemble methods which use different clustering algorithms to obtain base partitions in the ensemble generation step and to obtain a single partition in the consensus function that might create a compatibility issue in terms of working functionality in cluster ensemble architecture. Furthermore, the accuracy of the final results was a big worry to cope with. We tested our proposed framework on ten real-world benchmark datasets. The results showed that the proposed clustering ensemble method outperformed state-of-the-art clustering ensemble methods such as the CSPA, HGPA, MCLA, WSCE, EM, QMI, ECPCS MC, and ECPSCS HC algorithms on average. There are several strengths to the proposed cluster ensemble method; firstly, the same clustering functionalities in both of its stages lead the framework more compatible that significantly improves accuracy over state-of-art cluster ensemble methods. Second, it employs a newly proposed consensus function to combine base partitions into a single partition that uses information of cluster centers present in a data set to limit AP to produce a actual number of clusters rather than random number of clusters, resulting in a significant improvement in accuracy and execution time when compared to AP.

The proposed cluster ensemble method has several advantages that researchers can take advantage of. clustering is useful for extracting useful knowledge from large amounts of data. Cluster ensemble is the preferred option for reclustering previously obtained knowledge or hidden patterns from the clustering algorithm in knowledge reuse. The proposed cluster ensemble method can be used to reuse clustering algorithm knowledge and recluster it using the same clustering algorithm. As a result, it avoids the overheads associated with including another clustering algorithm for the consensus function.

As part of future work, we will further enhance the accuracy of CES and compare it to advanced cluster ensemble methods and datasets. We will optimise CES such that its time complexity will be comparable to other cluster ensemble methods. We will explore other cluster algorithms like AP features such as density peaks [32] that help in increasing accuracy significantly.

References

[1] Chang-Dong Wang, Jian-Huang Lai, and S Yu Philip. Multi-view clustering based on belief propagation. *IEEE Transactions on Knowledge and Data Engineering*, 28(4):1007–1021, 2015. doi: 10.1109/TKDE.2015.2503743.

[2] Cosmin Marian Poteraş, Marian Cristian Mihăescu, and Mihai Mocanu. An optimized version of the k-means clustering algorithm. In *2014 Federated Conference on Computer Science and Information Systems*, pages 695–699, 2014. doi: 10.15439/2014F258.

[3] Cosmin M. Poteraş and Mihai L. Mocanu. Evaluation of an optimized k-means algorithm based on real data. In *2016 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 831–835, 2016. doi: 10.15439/2016F231.

[4] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. doi: 10.1109/34.868688.

[5] Dimitrios Rafailidis and Petros Daras. The tfc model: Tensor factorization and tag clustering for item recommendation in social tagging systems. *IEEE Transactions on Systems, Man and Cybernetics:Systems*, 43(3):673–688, 2012. doi: 10.1109/TSMCA.2012.2208186.

[6] Dnyanesh G Rajpathak and Satnam Singh. An ontology-based text mining method to develop d-matrix from unstructured text. *IEEE Transactions on Systems, Man and Cybernetics: Systems*, 44(7):966–977, 2013. doi: 10.1109/TSMC.2013.2281963.

[7] Feiping Nie, Shaojun Shi, and Xuelong Li. Auto-weighted multi-view co-clustering via fast matrix factorization. *Pattern Recognition*, 102:107207, 2020. doi: 10.1016/j.patcog.2020.107207.

[8] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010. doi: 10.1016/j.patrec.2009.09.011.

[9] Ana LN Fred and Anil K Jain. Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):835–850, 2005. doi: 10.1109/TPAMI.2005.113.

[10] Pan Su, Changjing Shang, and Qiang Shen. A hierarchical fuzzy cluster ensemble approach and its application to big data clustering. *Journal of Intelligent & Fuzzy Systems*, 28(6):2409–2421, 2015. doi: 10.3233/IFS-141518.

[11] M. Yousefnezhad and D. Zhang. Weighted spectral cluster ensemble. In *2015 IEEE International Conference on Data Mining*, pages 549–558, Nov 2015.

[12] Tahani Alqurashi and Wenjia Wang. Clustering ensemble method. *International Journal of Machine Learning and Cybernetics*, 10(6):1227–1246, 2019. doi: 10.1007/s13042-017-0756-7.

[13] Alexander Topchy, Anil K Jain, and William Punch. Clustering ensembles: Models of consensus and weak partitions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1866–1881, 2005. doi: 10.1109/TPAMI.2005.237.

[14] Zhiwen Yu, Xianjun Zhu, Hau-San Wong, Jane You, Jun Zhang, and Guoqiang Han. Distribution-based cluster structure selection. *IEEE Transactions on Cybernetics*, 47(11):3554–3567, 2016. doi: 10.1109/TCYB.2016.2569529.

[15] Brendan J Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, 2007. doi: 10.1126/science.1136800.

[16] Kiri Wagstaff and Claire Cardie. Clustering with instance-level constraints. *AAAI/IAAI*, 1097:577–584, 2000. doi: 10.5555/645529.658275.

[17] Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3(Dec):583–617, 2002. doi: 10.1162/153244303321897735.

[18] D. Huang, C. Wang, H. Peng, J. Lai, and C. Kwoh. Enhanced ensemble clustering via fast propagation of cluster-wise similarities. *IEEE Transactions on Systems, Man and Cybernetics: Systems*, pages 1–13, 2018. 10.1109/TSMC.2018.2876202.

[19] Siavash Haghtalab, Petros Xanthopoulos, and Kaveh Madani. A robust unsupervised consensus control chart pattern recognition framework. *Expert Systems With Applications*, 42(19):6767–6776, 2015. doi: 10.1016/j.eswa.2015.04.069.

[20] Emmanuel Ramasso, Vincent Placet, and Mohamed Lamine Boubakar. Unsupervised consensus clustering of acoustic emission time-series for robust damage sequence estimation in composites. *IEEE Transactions on Instrumentation and Measurement*, 64(12):3297–3307. doi: 10.1109/TIM.2015.2450354.

[21] Tossapon Boongoen and Natthakan Iam-On. Cluster ensembles: A survey of approaches with recent extensions and applications. *Computer Science and Review*, 28:1–25, 2018. doi: 10.1016/j.cosrev.2018.01.003.

[22] Ashraf Mohammed Iqbal, Abidalrahman Moh'd, and Zahoor Khan. Semi-supervised clustering ensemble by voting. *arXiv preprint arXiv:1208.4138*, 2012.

[23] Teh Ying Wah Ali Seyed Shirkhorshidi, S. Aghabozorgi and Andrew R. Dalby. A comparison study on similarity and dissimilarity measures in clustering continuous data. *PLOS One*, 10:1–20, 2015. doi: 10.1371/journal.pone.0144059.

[24] Inmar Givoni and Brendan Frey. Semi-supervised affinity propagation with instance-level constraints. In *Artificial Intelligence and Statistics*, pages 161–168. doi: 10.1.1.158.678.

[25] Jie Hu, Tianrui Li, Hongjun Wang, and Hamido Fujita. Hierarchical cluster ensemble model based on knowledge granulation. *Knowledge-Based Sytems*, 91:179–188, 2016. doi: 10.1016/j.knosys.2015.10.006.

[26] Hongjun Wang, Hanhuai Shan, and Arindam Banerjee. Bayesian cluster ensembles. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 4(1):54–70, 2011. doi: 10.1002/sam.10098.

[27] Hongjun Wang, Jianhuai Qi, Weifan Zheng, and Mingwen Wang. Semi-supervised cluster ensemble based on binary similarity matrix. In *2010 2nd IEEE International Conference on Information Management and Engineering*, pages 251–254. IEEE, 2010. doi: 10.1109/ICIME.2010.5478054.

[28] Bo Liu, Hong-Jun Wang, Yan Yang, and Xiao-Chun Wang. The method of cluster ensemble based on minimum redundancy feature subset. In *Proceedings of the 2012 International Conference on Electronics, Communications and Control*, pages 2320–2323. IEEE Computer Society, 2012. doi: 10.5555/2417502.2418206.

[29] Zhi-Hua Zhou and Wei Tang. Clusterer ensemble. *Knowlwdge-Based Ssytems*, 19(1):77–83, 2006. doi: 10.1016/j.knosys.2005.11.003.

[30] Hao Li, Meng Wang, and Xian-Sheng Hua. Msra-mm 2.0: A large-scale web multimedia dataset. In *2009 IEEE International Conference on Data Mining Workshops*, pages 164–169. IEEE, 2009. doi: 10.1109/ICDMW.2009.46.

[31] Emrah Hancer. A new multi-objective differential evolution approach for simultaneous clustering and feature selection. *Engineering Application of Artificial Intelligence*, 87:103307, 2020. doi: 10.1016/j.engappai.2019.103307.

[32] Alex Rodriguez and Alessandro Laio. Clustering by fast search and find of density peaks. *Science*, 344(6191):1492–1496, 2014. doi: 10.1126/science.1242072.

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 137–140
DOI: 10.15439/2021F48
ISSN 2300-5963 ACSIS, Vol. 25

StarCraft strategy classification of a large human versus human game replay dataset

Štefan Krištofík, Peter Malík
Institute of Informatics, Slovak Academy of Sciences
Dúbravská cesta 9, 845 07 Bratislava, Slovakia
Email: stefan.kristofik@savba.sk

Matúš Kasáš
Tatramed Software
Líščie údolie 9, 841 01 Bratislava, Slovakia

Abstract—**Real-time strategy games are popular in AI research and education. Among them, Starcraft: Brood War (SCBW) is particularly well known. Recently, the largest known SCBW game replay dataset STARDATA was published. We classify player strategies used in the dataset for all 3 playable races and all 6 match-ups. We focus on early to mid-game strategies in matches less than 15 minutes long. By mapping the classified strategies to replay files, we label the files of the dataset and make the labeled dataset available.**

I. Introduction

IN a competitive one on one real-time strategy (RTS) game environment, players try to outsmart and defeat their opponent by using superior strategy. This involves planning and execution of various tasks like good army composition, military unit placement, effective combat, scouting, territory expansion. RTS are considered very challenging for AI. The main reasons are partial observability of the game state (visible only near own units and structures) and huge complexity resulting from an overwhelming number of possible player actions [2].

StarCraft: Brood War (SCBW) released in 1998 (Fig. 1) is the most successful [9] and widely known RTS game in both the competitive and AI research communities. The competitive side has been praising the unique and fair balance [8] of all 3 playable races (Protoss, Terran, Zerg) which is rarely accomplished in gaming and is one of the main reasons for the game's longevity. The research side benefits from this longevity and over the years many useful tools were developed to help with AI research. Current goal of ongoing research is the continuous improvement of agents and ultimately overcoming human expert players on a consistent basis, which they are not yet capable of as evidenced by the results of recent AI game competitions [3], [4], [9].

One of possible approaches to agent improvement is machine learning from the past matches. SCBW allows archiving of played matches in the form of replay files. Over the years, a vast amount of such data was accumulated. However, it is scattered among many sources with various levels of quality. In case of SCBW, for machine learning purposes a dataset of game replay files should meet multiple requirements to be considered viable, mainly diversity, universality and validity [1], [8]. Recently, a high quality SCBW replay dataset called STARDATA which adheres to these requirements was published by Facebook [1]. It is a collection of 65646 game replays and is the largest dataset compiled to date. Other smaller datasets were introduced previously [5], [6], [7].

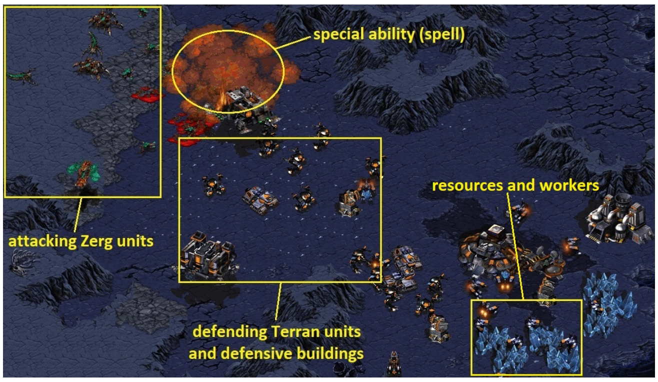


Fig. 1. StarCraft: Brood War

Following are the main contributions of this work. We classify strategies used by both players in each match from STARDATA and label the replay files with identified strategies. We make the labeled dataset publicly available [1].

II. Strategy classification

This work deals with strategy classification from the SCBW replay dataset STARDATA. Strategy classification was identified as one of the tasks which the dataset is suitable for [1]. Having such information could be helpful for future machine learning attempts and improvement of SCBW AI agents.

Some attempts at strategy classification from SCBW replays were conducted [5], [6], [7], [1], [8], but the results of each work were quite limited or very specific in some way (dataset size, quality, number of strategies, races).

In this work, we expand the idea of [8] where strategies were extracted from STARDATA, but only for the Terran race. We now consider and classify strategies of all 3 playable races in all 6 match-ups. We provide strategy information in the form of labeled files. The classification and labeling process is shown in Fig. 2 and described next.

This work is supported by Slovak national project VEGA 2/0155/19.

[1] https://drive.google.com/file/d/1mpQZoNN51iv7UX-IBanRsz0756AKO9Y3

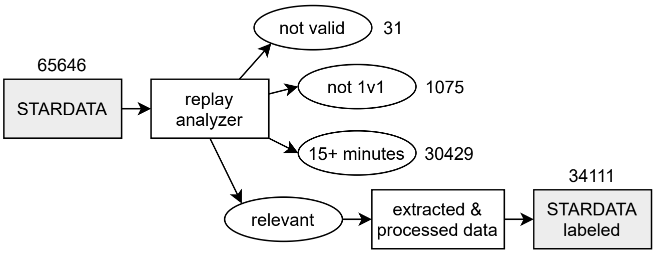


Fig. 2. Strategy classification and labeling. Numbers=amounts of replay files

TABLE I
SELECTED STRATEGY DEFINING STRUCTURES AND UNITS

Race	Structures	Units
Terran	Academy, Armory, Comm. Center, Com. Station, Con. Tower, Eng. Bay, Factory, Machine Shop, Science Fac., Starport, Refinery	Marine, Medic, Firebat, Vulture, Goliath, Siege Tank, Wraith
Protoss	Nexus, Gateway, Forge, Stargate, Cyber. Core, Templar Arch., Stargate, Robotics Facility	Zealot, Dragoon, High Tem., Dark Tem., Carrier
Zerg	Hatchery, Lair, Spire, Spawning Pool, Hydralisk Den	Zergling, Lurker, Scourge Hydralisk, Mutalisk

A. *Dataset cleaning and filtering*

In this work, we are interested in valid competitive 1v1 matches and early to mid-game strategies. For the match length threshold, we chose 15 minutes for the same reasons as described in [8].

We use our own software tool called BWAPI replay analyzer to automatically process the original replay files of STARDATA. It utilizes BWAPI [2], an open source API for SCBW to interact with the game engine and play back replay files, one by one, gathering useful information from each match.

1) Validation: First, each replay is checked for validity; if it can run correctly in BWAPI. Invalid replay files are removed from the relevant file pool and excluded from further processing, thus cleaning the dataset of them.

2) Filtering: Next, if a replay is valid it is checked for competitiveness and length. Only matches less than 15 minutes long and including 2 players are kept in the relevant file pool.

Dataset cleaning and filtering results are visible in Fig. 2. 34111 replays (52 % of the original dataset) remained in the pool of replays relevant for strategy classification.

B. *Raw information extraction*

To extract raw detailed information from relevant replay files and store in into json files (one json for each replay file), we created a modified version of the replay extractor for the Terran race introduced in [8]. The tool can now extract information from all 6 match-ups, automatically categorize it by match-up and map it to replay files.

C. *Information processing*

We further process the raw information and prepare it for strategy classification. SCBW strategies can be characterized mainly by [8] a) build orders - sequences of structure construction, and b) army compositions - lists of backbone unit types, i.e., the most used unit types. To collect the information about a) and b), we extract the following from each json file:

- Basic info: file name, player names and races, match length, map name, winner [8].
- For both players: count of all structure types.
- For both players: relative count of all unit types per minute. This is computed as total count divided by match length in minutes.
- For each structure and unit type: timestamp of the first occurrence.

The information is consolidated into 6 csv files, one for each match-up, each storing one match per line.

D. *Strategy classification*

We classify strategies of all 3 races in all 6 match-ups. Because each race would use different set of strategies against different opponents, we divide strategies into 9 categories: PP, PT, PZ (for Protoss), TT, TP, TZ (for Terran), ZZ, ZP, ZT (for Zerg). For example, Protoss would use different set of strategies against Terran (PT) than against Zerg (PZ).

Based on our domain knowledge and experience with the game, we have selected a set of most important structures and units from each race's repertoire which will be used to define various strategies. The list is shown in Table I.

Let s be the number of selected structure types in Table I. Considering only structures listed in Table I, we compute for each match in each csv file the build order as follows:

- Assign value 1 to the first structure type to be constructed by a player during a match.
- Assign value 2 to the second, 3 to third, etc., up to s.
- Assign value $s+1$ to all structure types never constructed during a match.

Let u be the number of selected unit types in Table I. Considering only units listed in Table I, we compute for each match in each csv file the unit frequency statistics as follows:

- Assign values from 1 up to u depending on the relative frequency during a match. Unit types created with higher frequency get lower values and vice versa.
- Assign value $u+1$ to all unit types never created during a match.

For strategy classification, we treat STARDATA as unlabeled data because strategies used by both opponents are unknown. We perform classification by the K-Means clustering algorithm. The goal is to identify regularities in the data. By grouping similar data into clusters, we can differentiate between various strategies. Each cluster will represent a distinct strategy. After careful adjusting and result inspection, we chose 10 as the target number of clusters for the algorithm per strategy category. This guaranteed sufficient diversity of clusters and also sufficient abundance of replays per cluster. The algorithm produces differently sized clusters. The more

[2] https://github.com/bwapi/bwapi

popular a strategy is the larger the cluster representing it will be. The outcome is a total of 90 identified strategies, 30 for each race, 10 per category.

III. Results

A small sample of classified strategies is shown in Fig. 3. Complete results with all 90 strategies are available [3]. Strategy distributions are shown in Fig. 4.

A. Strategy descriptions

Strategies in Fig. 3 are named based on categories listed in II-D and the cluster number assigned by the K-Means algorithm. For example, TP6 is a Terran player strategy used against Protoss with the cluster number 6 assigned by K-Means. The amounts of players that used each strategy are shown in column *count*. Columns *average structure order* show the build order values (explained in II-D) averaged over all the matches in each cluster. Columns *average unit frequency* show the unit frequency values (explained in II-D) averaged over all the matches in each cluster. A short verbal description of strategies is given in the last column. Descriptions focus on different aspects of strategies. In general, we are interested in the following information about each strategy:

1) Most used units: Examples: In strategy PP9, *DZ* means the most used units are Dragoons and Zealots. In ZT7, *M* means the most used unit is Mutalisk.

2) Other used units: Example: In strategy TT4, *often WMV* means Wraiths, Marines and Vultures are used very often, but are not the most used units.

3) Significant structures: If a player has built some particular structures, it may indicate they are going to produce some specific units excluded from the list in Table I. Examples: Robotics Facility built by Protoss players in strategy PP0, Lair built by Zerg players in strategy ZP6.

4) Economic expansion strategies: These try to expand economically very early and gain an income advantage over the opponent. Examples: *fast exp* indicates this type of strategy, *late exp* does not indicate it.

5) Rush strategies: These try to end the match as soon as possible by attacking or pressuring the opponent very early and catching them unprepared. Examples: *Cannon rush* in strategy PZ3, *Z rush* in ZP1.

B. Discussion

The results for each of 3 races show good variety among identified strategies. Not only 'normal' strategies are represented, but also some rush as well as economic strategies are in the mix. However, it also indicates that some tweaking of strategy defining features might be needed. For example, Carriers for Protoss are almost never used in any strategy, but Spawning Pools for Zerg are almost always built.

The results in Fig. 4 prove the variety of identified strategies for all 3 races is good which was achieved by correct selection of clustering parameters. For each race and each match-up, one can clearly identify few favorite strategies (e.g., PP0, TZ7, ZZ4) as well as those less popular (e.g., PT6, TZ0, ZZ8).

IV. Dataset labeling

We label the original STARDATA replay files by adding the following information:

- Strategies for both players.
- Match-up (can be inferred from the strategies).
- Winner flag.

Example original file: *bwrep_0xi84.rep*.

Labeled file: *bwrep_0xi84_TZ7_ZT2W.rep*.

The original unique replay ID number is preserved. Player 1 was Terran and used strategy TZ7. Player 2 was Zerg and used strategy ZT2. The winner was Player 2, indicated by the symbol *W* after their strategy.

V. Conclusion

We classify player strategies used in StarCraft: Brood War replay files from the largest known unlabeled dataset called STARDATA and label the replay files. The replay files in the labeled version now offer information about match-up and strategies used by both players and also identify the winning player. While original STARDATA may be used for unsupervised learning, in machine learning, it is always beneficial to have more options. Having the above information available makes the labeled version useful for supervised learning. We make the labeled dataset available for future machine learning attempts for StarCraft AI agent training and improvement.

References

[1] Z. Lin, J. Gehring, V. Khalidov and G. Synnaeve, "STARDATA: A StarCraft AI Research Dataset," *13th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, AIIDE 2017,* pp. 50–56, arXiv:1708.02139.

[2] S. Ontañon, G. Synnaeve, A. Uriarte, F. Richoux, D. Churchill and M. Preuss, "A Survey of Real-Time Strategy Game AI Research and Competition in StarCraft," *IEEE Transactions on Computational Intelligence and AI in games, IEEE Computational Intelligence Society, 2013,* 5(4), pp. 1–19, doi: 10.1109/TCIAIG.2013.2286295.

[3] Mi. Čertický, D. Churchill, K.-J. Kim, Ma. Čertický and R. Kelly, "StarCraft AI Competitions, Bots and Tournament Manager Software," *IEEE Transaction on Games, 2018,* 11(3), pp. 227–237, doi: 10.1109/TG.2018.2883499.

[4] O. Vinyals, I. Babuschkin et al., "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature, 2019,* 575, pp. 350–354, doi: 10.1038/s41586-019-1724-z.

[5] B. G. Weber and M. Mateas, "A data mining approach to strategy prediction," *IEEE Symposium on Computational Intelligence and Games, 2009,* pp. 140-147, doi: 10.1109/CIG.2009.5286483.

[6] H. C. Cho, K. J. Kim and S. B. Cho, "Replay-based strategy prediction and build order adaptation for StarCraft AI bots," *IEEE Conference on Computational Intelligence in Games (CIG), 2013,* pp. 1-7, doi: 10.1109/CIG.2013.6633666.

[7] G. Synnaeve and P. Bessière, "A Dataset for StarCraft AI & an Example of Armies Clustering," *Artificial Intelligence in Adversarial Real-Time Games, 2012,* arXiv:1211.4552.

[8] Š. Krištofík, P. Malík, M. Kasáš, Š. Neupauer, "StarCraft agent strategic training on a large human versus human game replay dataset," *Federated Conference on Computer Science and Information Systems, FedCSIS 2020,* 21, ACSIS, pp. 391–399, doi: 10.15439/2020F178.

[9] M. Świechowski, "Game AI Competitions: Motivation for the Imitation Game-Playing Competition," *Federated Conference on Computer Science and Information Systems, FedCSIS 2020,* 21, ACSIS, pp. 155–160, doi: 10.15439/2020F126.

[3] https://drive.google.com/file/d/184ZV5VCzj75avTxjxSM1qLncRjmzs_9o

		average structure order (1=first, 8=never)							average unit frequency (1=always, 6=never)					
Protoss strategy	count	Nexus (exp)	Cybernetics Core	Gateway	Forge (F)	Templar Archives	Stargate (S)	Robotics Facility (R)	Dragoon (D)	Zealot (Z)	High Templar (HT)	Dark Templar (DT)	Carrier (C)	brief description
PP0	2222	3,85	2,01	1,01	8,00	7,82	7,99	3,14	1,03	2,21	6,00	6,00	6,00	DZ, fast R, exp
PP9	1001	4,71	2,06	1,00	4,27	7,56	7,98	3,53	1,13	2,05	5,97	5,99	6,00	DZ, R, F, exp
PT1	427	4,76	2,02	1,00	7,40	3,80	7,72	3,99	1,58	3,79	5,90	1,63	5,97	DT rush, often D, exp
PT3	424	8,00	2,00	1,00	7,87	7,93	7,97	3,01	1,09	3,15	6,00	6,00	5,99	D, often Z, fast R, no exp
PZ3	586	3,37	7,94	4,11	1,37	8,00	8,00	8,00	6,00	4,14	6,00	6,00	6,00	Cannon rush, exp
PZ9	388	5,79	2,14	1,03	3,94	3,63	7,97	6,93	3,37	1,34	3,74	3,65	6,00	Z, often D HT DT, late exp

		average structure order (1=first, 12=never)											average unit frequency (1=always, 8=never)							
Terran strategy	count	Academy (A)	Armory	Command Center (exp)	Comsat Station	Control Tower	Engineering Bay	Factory	Machine Shop	Science Facility	Starport	Refinery	Marine (M)	Vulture (V)	Goliath (G)	Siege Tank (S)	Wraith (W)	Medic (E)	Firebat (F)	brief description
TT4	241	6,36	10,17	4,95	8,74	6,78	10,46	2,22	4,10	11,89	4,08	1,21	3,28	4,18	7,90	1,83	2,54	7,94	7,92	S, often WMV, exp
TT6	211	6,26	5,00	4,43	8,11	11,27	6,45	2,09	3,50	11,83	10,09	1,08	3,67	3,63	2,31	2,02	7,33	7,98	7,97	GS, often MV, few W, exp
TP5	220	11,35	12,00	10,27	12,00	12,00	11,73	12,00	12,00	12,00	12,00	8,79	1,83	8,00	8,00	8,00	8,00	7,95	7,94	M rush, late exp
TP6	731	11,59	11,89	12,00	12,00	12,00	8,83	2,01	3,02	12,00	11,59	1,00	2,30	2,17	7,95	2,65	7,89	7,99	8,00	MVS, no exp
TZ3	917	8,79	4,96	4,45	10,06	10,36	6,91	2,31	4,17	11,45	9,21	1,26	3,26	3,09	1,80	4,09	7,57	7,54	7,95	G, often MVS, few WE, exp
TZ7	2005	3,18	11,88	1,30	4,80	9,44	4,20	5,75	7,39	10,01	7,77	1,94	1,06	7,11	7,95	2,70	7,70	2,83	5,42	M, often SEF, few V, fast exp

		average structure order (1=first, 6=never)					average unit frequency (1=always, 6=never)					
Zerg strategy	count	Hatchery (exp)	Lair (A)	Spawning Pool	Hydralisk Den	Spire	Zergling (Z)	Hydralisk (H)	Lurker (L)	Mutalisk (M)	Scourge (S)	brief description
ZZ3	1579	3,97	2,00	1,00	6,00	3,16	2,66	5,99	6,00	1,82	1,54	MS, often Z, fast A, exp
ZZ7	959	1,82	2,79	1,39	5,99	4,00	1,48	5,99	5,98	1,55	6,00	ZM, A, fast exp
ZP1	527	1,72	3,02	1,32	5,58	4,87	1,08	6,00	6,00	6,00	5,05	Z rush, few S, A, fast exp
ZP6	826	1,73	3,41	1,33	3,56	5,68	2,12	1,92	2,16	5,82	5,93	ZHL, A, fast exp
ZT2	1027	1,31	3,00	1,71	4,82	4,15	2,02	4,07	1,61	2,72	5,75	LZ, often M, few H, A, fast exp
ZT7	451	1,42	2,98	1,64	5,47	4,13	2,16	5,61	5,86	1,97	2,11	M, often ZS, A, fast exp

Fig. 3. A small sample of Protoss (top), Terran (middle) and Zerg (bottom) strategies

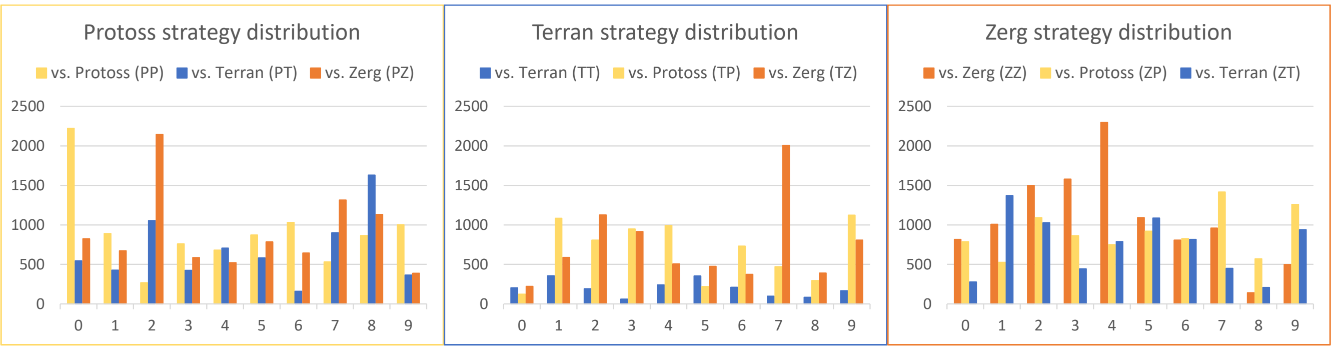


Fig. 4. Strategy distribution

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 141–145
DOI: 10.15439/2021F130
ISSN 2300-5963 ACSIS, Vol. 25

Pix2Trips – a system supporting small groups of urban tourists

Halina Kwasnicka
Wroclaw University of Science and Technology
wyb. Wyspianskiego 27, 50-370 Wroclaw, Poland
Email: halina.kwasnicka@pwr.edu.pl

Tsvetan Ovedenski
Chaos Group
Bulgaria
Email: tsvetan@tsovedenski.com, ccovedenski@gmail.com

***Abstract*—Group recommendation systems are the subject of many publications, but still is a gap between research results and group decision support systems' needs. Tourists often do not know which attractions they would like to visit. The *Pix2Trips* system asks the group's members to indicate images that they would like. On this basis, *Pix2Trips* models the group's preferences and adjusts them to the proposed places' models. Some tourist places in Wroclaw city, Poland, were used in experiments. The paper presents the system's components and discusses the results of the experiments. Conclusions indicate the good overall evaluation of the *Pix2Trips* system and further research.**

I. Problem statement

LET us imagine that we want to visit monuments in a city in a group of several people. We need to make a list of places to visit so that all group members are reasonably satisfied. The decision about the places worth visiting requires a meeting and discussion within the group, but we can unknown all potentially interesting places in a new city. We aim to develop a computer system targeting groups of city tourists which should enable remote cooperation. In the proposed system, users define their preferences only by selecting multiple images. It means that the entire group's recommendation is defined based on the individual preferences indicated by chosen images. Our goal is to check if this method combined with an aggregation strategy will work for small groups. The research question is: how different aggregation strategies, including two proposed by authors, influence the quality of recommendations? Section 2 presents an overview of Group Recommender Systems. Section 3 describes the proposed Pix2Trips system. In the 4th section, we briefly discuss experiments with artificial and real groups. Conclusions end the paper.

II. Group Recommender Systems (GRS)

Group Decision Support Systems (GDSS) extend Decision Support Systems (DSSs) with groupware functionality and various combination preferences strategies. The popular classification of groupware considers group proximity and time of communication [1]. The system can be targeted at groups located in:

- same place and same time (difficult in larger groups [2]),
- same place and different time,
- different place and the same time,
- different place and different time.

Expanding recommended systems to support multiple users requires that either profiles or recommendations be merged [3]. Eight different strategies are presented in [4]:

1) *Average* – averages individual values,
2) *Average without misery* – averages values below a threshold,
3) *Multiplicative* – multiplies individual values,
4) *Least misery* – takes the minimum of individual values,
5) *Most pleasure* – takes the maximum of individual values,
6) *Approval voting* – counts values above a threshold,
7) *Borda counts* – assigns positions to values and sums them up,
8) *Dictatorship* – takes values of the most respected individual.

The effect of group dynamics is usually ignored in literature [5]. To solve the *cold start problem*, i.e., when we do not have information about a new user, two methods are used: one traditional using the questionnaire and the second – picture-based. We focus on the second one. Authors of [6] proposed picture-based elicitation where users select pictures as a base for profile determining. They use a Seven-Factor model based on the analysis Big Five personality factors [7], and 17 tourist roles [8] representing short-term preferences. The seven factors are the following: (1)sun-loving and connected, (2)educational, (3)independent, (4)culture-loving, (5)open-minded and sportive, (6)risk-seeking, (7)nature and recreation. The needed data is gathered in two steps: (1)each picture is assigned one factor as the most relevant, (2)the recommendation base is built manually, with experts' help. Authors of [9] tried to automate this tedious and subjective task via text-mining techniques. Paper [10] proposes another way of mapping images to tourist types: they defined 17 tourist roles and the tourism-related images. In [11] a user selects photos in several rounds: in subsequent rounds, presented pictures are refined based on the previous choices. The functionality of Groupware is connected with the size and proximity of target groups. INTRIGUE [13] targets large heterogeneous groups of unfamiliar people. In contrast, STSGroup [14] and Hootle+ [15] are more suitable for small groups of family or friends.

III. *Pix2Trips* – the proposed system

The proposed system is dedicated to small groups of tourists to suggest interesting places in a new city. Users do not complete any questionnaire or provide personal data. The system asks the user to indicate which objects from pictures presented to him he would like to visit. On this basis, the system generates the user profile. Fig. 1 presents a general scheme of *Pix2Trips*. *Pix2Trips* is a web application working on recent versions of browsers on a PC or laptop. Users do not need credentials, their identity is generated on the first visit and is kept locally (*Authentication*). A user is

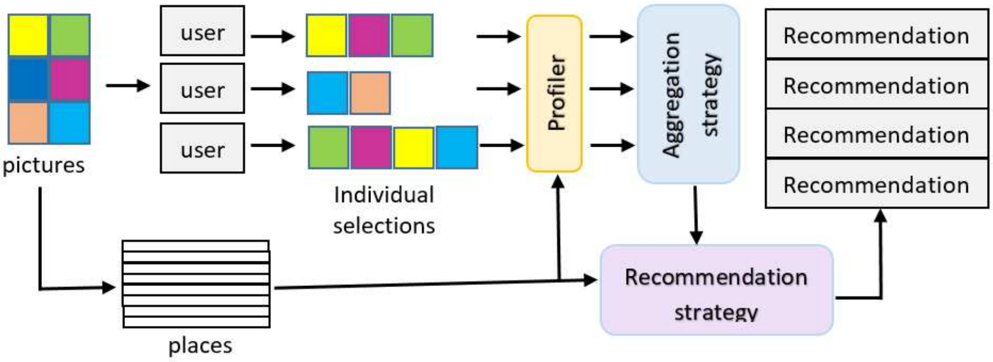


Fig. 1. A general scheme of the *Pix2Trips* system

represented by a name. *Sessions*: a session is a combination of users, individual picture selections and selected places. The group initiator creates a session, the group members join the session by a hyperlink or a unique 6-digit auto-generated code. *Voting for suggestions*: suggestions are derived from the users' picture selections, they are saved for later reference. *Real-time interaction*: users of one group belonging to one session should see each others interaction in real-time.

Specifications. To give a group of urban tourists recommendations, we need a consistent model for group preferences and places. Both have to be represented by the same set of attributes. The set of attributes depends on the tourism specificity of a given city or region (e.g., a beach). Our example is Wroclaw city. We have defined a list of 44 features used for modeling places for the recommendation. Each feature is assigned to each place with an appropriate intensity value, in scope [0, 100], where 0 means irrelevant feature, and 100 – the highest value. We used 39 pictures taken from a freely available popular Internet resource Unsplash at https://unsplash.com, none of them was taken in Wroclaw. The base of places for recommendation consists of 60 entries, all of them are located in Wroclaw. All places can be grouped into 11 subjects providing 44 features. The selected pictures present different objects: Museum, Sight, Church, Restaurant, etc. The list of 11 types of places and 44 features is following:

1) *Activity*: cultural, eating, games, party, shopping, walking
2) *Architecture*: baroque, gothic, modern, neoclassic, renaissance
3) *Art*: classic, exhibitions, modern, street
4) *Geo*: beach, lake, mountains, river, urban
5) *Food*: burgers, french-fries, pizza, polish, quick-meal, steaks
6) *Drink*: beer, beer-craft, cocktails, shots, wine
7) *Location*: indoor, outdoor
8) *Subject*: children, family, elderly
9) *Related*: history, religious, science, transport
10) *Others*: bridge, scenic-lookout
11) *Has*: animals, greenery

Aggregation strategies. In *Pix2Trips*, we implemented nine aggregation strategies, seven from literature [12], and two ours: *Average + Normalize*, and *Composite*. Below we present all aggregation methods. In all formulas, f_i is an aggregated value of the *i*th feature in the group; N is the size of the group; M is the number of features; i is the feature number; x_{ki} is a value of *i*th feature of the *k*th member of the group.

Average: it is the most straightforward and most natural strategy. Each factor is averaged between all members.

Average + Normalize: each value is normalized to 100 so the factors with highest values become the most important.

$$f_i = 100 \times \frac{\sum_{k=1}^{N} x_{ki}/N}{\max_{j\in[1;M]} \sum_{k=1}^{N} x_{kj}/N} \tag{1}$$

Average without Misery: it is a standard *average* strategy in which values below a certain threshold are removed (set to zero). We set the threshold at 33%.

Multiplicative: the values of each feature are multiplied. The strategy eliminates features unwanted by even one person.

$$f_i = 100 \times \frac{\prod_{k=1}^{N} x_{ki}}{\max_{j\in[1;M]} \prod_{k=1}^{N} x_{kj}} \tag{2}$$

Least Misery: it focuses on minimizing the overall preferences, i.e. assumes that the group is as satisfied as to the least happy member.

$$f_i = \min_{k\in[1;N]} x_{ki} \,|x_{ki} > 0 \tag{3}$$

Most Pleasure: it acts as the opposite of *Least Misery* strategy, i.e., it prefers the highest values of the features.

$$f_i = \max_{k\in[1;N]} x_{ki} \tag{4}$$

Approval Voting: here, values greater than or equal to the assumed threshold are replaced by value 100, the rest is changed to zero. Next, the *Average* method is applied to modified vectors. We used thresholds equal to 33% and 50%.

Borda Count: it uses the ranks of values ordered in ascending order. If multiple features have the same value, they are assigned the same rank, but the next factor skips as many ranks as duplicates. Next, values are summed, and the resulting vector is scaled according to the maximal possible value.

Composite: it represents the ability to combine other strategies. Multiple group models are created using different strategies, next they are combined again. In *Pix2Trips* we implemented *Multi1* strategy – a combination of *Least Misery* and *Most Pleasure* by *Average*.

Image-based preferences and recommendation strategies. The idea is based on [6], in which the preferences represent relevance to each of the seven touristic factors, but we use a different set of features. To compute the relationship between the pictures and the features, we apply the multivariate linear regression for each feature f_i:

$$f_i = \sum_{j=1}^{P} b_{ij} x_j, \quad \text{for } i \in [1, M] \tag{5}$$

where P is a number of pictures, and M is a number of features (in *Pix2Trip* $P = 39, M = 44$).

We have to calculate b_{ij} to determine the user's profile out of pictures. f_i is the known numerical value representing the relevance of the feature f_i to the place under consideration. Value of x_j is calculated in three ways (types) [6]. *Type 1*: x_j is assigned 1 if the picture has been selected and 0 otherwise. It does not use the positional information of the assigned images. *Type 2*: the value of x_j for the first picture is 1, for the 2nd is $(M-1)/M$, according to the formula: $x_j = (-k+M+1)/M$ if picture j is selected on kth position, and 0 otherwise. *Type 3*: the positional information k and the number of assigned pictures n are considered. $x_j = M\frac{(-k+n+1)}{\sum_{m=1}^{n} m}$ if picture j is selected on kth position out of n, and 0 otherwise. Once calculation the model of each place, we can determine the *user profile* x_j^u based on the images he chose and one of the above methods (*type1*, *type 2* or *type 3*)

Recommendation strategies. We can calculate the distance between the group's profile and all available places because both places and aggregated users' profiles are the M-dimensional space points: $d : \Re^M \times \Re^M \rightarrow \Re$. Vectors describing places are relatively sparse. Different distance measures can be used, in *Pix2Trips*, we implemented *Euclidean*, *Manhattan* and *Chebyshev*. In *Pix2Trips*, three recommendation strategies were implemented: (1)*All*: full-length vectors are used, (2)*Non-zero*: only features with strictly positive values are considered, (3)*Top N*: profile's N most desired features are taken, regardless of venue

When the group model and the place description are incompatible, the distance is measured between empty vectors. In such situations, the distance is set as a constant, sufficiently large value.

IV. Experimental study

Pix2Trips is accessible at https://pix2trips.xyz/. Experiments were conducted on a laptop with: CPU: Intel i7-3630QM; RAM: 32GB; OS: Arch Linux5 (5.4.25-2-lts Linux kernel). Implementation of algorithms is done in Kotlin programming language (version 1.3.71), and the code is run on Java Virtual Machine 11.0.6 (OpenJDK7). The used parameters influence a set of recommended items. Each item is characterized by distance, i.e., how relevant a place is to the group profile. We performed a series of experiments to determine the best combination of system parameters. The output list of distances is aggregated into min, max, average and standard deviation to compare different parameters. In all experiments we have used a dataset consisting of 60 places in Wroclaw city, represented by ID number. Due to the limited space, we do not provide details of these experiments, only a brief overview of them and their results are mentioned. In more detail, we discuss the experiments regarding the quality of the system's recommendations, made with real participants of the study.

A. Experiments with artificial groups with different characteristics

We used three simulated groups with different degrees of shared interests (Table I). Members of group 1 have mostly common interests, group 2 has only some common interests, and group 3 have no common interests. Surprisingly, aggregation strategy *Average* + *Normalize* has the lowest distance regardless of the group. The difference in groups is visible in the *Multiplicative* strategy, where no commonality resulted in an empty recommendation list. *Approval Voting* 50% performed poorly on groups with common interests, only one recommendation was produced with this strategy for group 1 and zero for group 2. Strategy *Multi1*, which is a *composite* strategy, has shown little variance between the groups. Almost all strategies took comparable time to calculate the group model ($\approx$ 27 ms.). The only outlier is *Borda Count*, as the algorithm for this strategy is more sophisticated ($\approx$ 30 ms.). *Average+Normalize* performs almost the same regardless of the recommendation strategy. The *Multiplicative* strategy explicitly shows what effect each strategy has: the higher distance is with strategies *All*, a bit smaller with *Top 40*. Also, *Non-Zero* produces a relatively high distance. Strategy *Average+Normalize* does not show the effect of different profile strategies. *Average* aggregation strategy shows that *Type1* and *Type2* profiles result in recommendations that are closer to the group model. *Type3* considers the assigned images, their order and the total count, resulted in the considerably higher distance between recommended places and the group profile. Generally, *Average+Normalize* aggregation strategy had the lowest distance, which corresponds to recommendations closest to the group profile. Changing other parameters and using the different characteristics of the artificial groups did not significantly influence the results. Based on the experiments, we can predict that configurations *Average+Normalize* aggregation strategy, *Type1* profile strategy and either *Non-zero* or *Top 1* recommendation strategy would perform better than the others in terms of accuracy.

B. Pix2Trips evaluation by users

We asked members of seven groups of 3 or 4 members each (8 females, 14 males) – computer science students and people between 35 and 55 years old with moderate technical knowledge, to evaluate *Pix2Trips*. They had no prior knowledge about *Pix2Trips* and were familiar with the touristic places in Wroclaw to evaluate each recommendations list properly. We have used the *System Usability Scale* (SUS), which is a set of ten questions aiming to estimate the overall usability of a system [17]. Users assign a value between 1 (*strongly disagree*) and 5 (*strongly agree*) – known as the *Likert scale*. The questions are formulated alternately: positive and negative. Assigned values are adjusted that low value always corresponds to negative answer and high value – to positive. The 10 SUS questions are the following:

- SUS1. I think that I would like to use this system frequently.
- SUS2. I found the system unnecessarily complex.
- SUS3. I thought the system was easy to use.
- SUS4. I think that I would need the support of a technical person to be able to use this system.

TABLE I
PICTURE SELECTION BY THE MEMBERS OF THREE ARTIFICIAL GROUPS

#	Picture selection: Group 1	Picture selection: Group 2	Picture selection: Group 3
1	4, 34, 22, 35, 11, 6, 13, 25, 31	4, 34, 22, 11, 6, 13, 25, 31, 35	7, 36, 2, 22, 6, 21, 3, 9, 34
2	6, 2, 33, 30, 17, 8, 23, 39, 28	2, 33, 23, 28, 27, 1, 26, 7, 21	30, 4, 35, 28, 29, 19, 20
3	35, 37, 39, 8, 40, 17, 3, 24, 13, 4	37, 8, 40, 17, 3, 5, 38, 2, 22	11, 18, 6, 12, 17, 32, 13, 8
4	32, 26, 19, 40, 34, 30, 23, 3, 4, 12	32, 40, 34, 30, 23, 3, 2, 9, 36	24, 31, 14, 16, 37, 26, 33

- SUS5. I found the various functions in this system were well integrated.
- SUS6. I thought there was too much inconsistency in this system.
- SUS7. I would imagine that most people would learn to use this system very quickly.
- SUS8. I found the system very awkward to use.
- SUS9. I felt very confident using the system.
- SUS10. I needed to learn a lot of things before I could get going with this system.

According to [18], a mean score for web applications is 68 (on a scale of 1 to 100) – this value is our baseline for comparing our application to others with the same user interface type. Authors of [19] propose *ResQue* – an evaluation framework containing various questions/statements aiming to quantify different aspects of a recommender system from the end-users point of view. We focus on two aspects: the quality of recommendations and the enjoyability of the preference elicitation process. We divided the *Pix2trips* evaluation into two stages: in-app, and post-experiment. In the first stage, session members evaluated the system usability, i.e., the resulting list of items showing perceived accuracy on a seven point scale: *Worst*, *Awful*, *Poor*, *Fair*, *Good*, *Excellent*, *Best*. The second part was after the group experiment. Evaluators filled out a questionnaire consisting of 10 SUS statements, six different categories questions adopted from *ResQue* framework, and three questions concerning group identity [20]. A list of questions contained in the questionnaire is following:

- The 10 SUS questions
- Qual1. The recommender gave good suggestions.
- Qual2. The recommended items are diverse.
- Qual3. I found it easy to tell the system about my preferences.
- Qual4. The recommended items took my preferences into account.
- Qual5. Finding places to visit with the help of the recommender is easy.
- Qual6. If a recommender such as this one existed for other cities, I would use it to find places to visit.
- Identity1. I am happy to be a part of this group.
- Identity2. I consider my preferences similar to the rest of the group.
- Identity3. I feel the group has acted as a team.

The below scenario was given to the evaluators.

"Imagine that you, together with your friends, are planning a trip to Wroclaw. You have two full days in which you can visit and explore different places. First, you should individually choose up to 10 pictures that you most like and identify with them. After that, discuss by chat the recommendations with your group members and try to choose particular places to visit."

In questionnaires, we use a *Likert* scale. Following [21], we decided to use a *7-point scale* in our statements, where, depending on the nature of the question, lowest possible (equal to 1) is labeled as *worst imaginable* / *strongly disagree*, middle point (4) is labeled as *fair* / *neither*, and the highest label (7) is *best imaginable* / *strongly agree*. The SUS scores were adjusted: from odd-numbered questions we subtracted 1, answers of even-numbered questions were subtracted from 7. The answers were added together and scaled to 100.

All individuals score the system usability higher than the baseline (Fig. 2). The overall score for *Pix2Trips* is 85, which is in the 4*th* quartile. The highest average scores were obtained for SUS3 (*Pix2Trips* is uncomplicated), and SUS7 (is easy for new users). SUS2 and SUS10 scored the lowest, i.e., the system and its user interface are not complex to use. Recommendations quality received overall positive feedback.

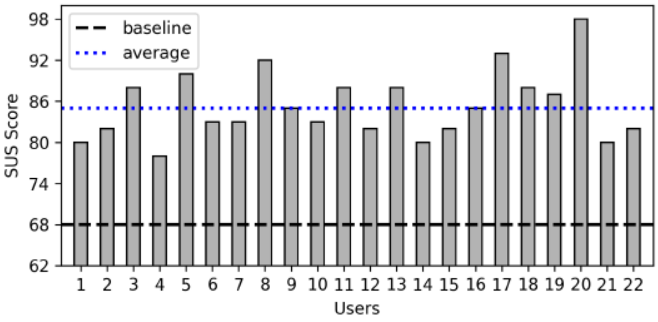


Fig. 2. SUS score per user

The users agreed that *Pix2Trips* generated good suggestions for their group (Qual1). Most of the evaluators assessed recommendations diversity (Qual2) as proper. 95% of users found the specification of their preferences via pictures as easy (Qual3). According to free-form feedback, the elicitation process was often labeled as "interesting". All users found finding new places (Qual5) as easy and would use *Pix2Trips* for other cities (Qual6). Members of each group were happy to be together (Identity1). There was a large variance of the perceived similarity in preferences among the group members (Identity2) with 10% seeing as they have different preferences than the rest of the group. This explains the indecisive answers for Qual4, on which 41% of users put the highest score. 86% of the respondents agree that the team has acted as a team (Identity3). Users reckon that the system is beneficial for new tourists, but it could be equally useful for residents.

The in-app assessment gave us more than 350 ratings that included recommendations with different preferences and parameters, making it possible to check the correlation between distance measure and user ratings. Most ratings are around the midpoint, with the most common rating of 5 (36.1%). Spearman's rank correlation coefficient $\rho = -0.110$ indicates a weak negative correlation between average distance and rating. It suggests that the distance measure is not enough to be used as a metric for recommendations quality. Most of the ratings are between midpoint (i.e., 4) and 5. Aggregation strategies *Most Pleasure* and *Average without Misery* are highest rated – 4.86 and 4.85, respectively. The classic *Average* was rated moderately (4.63) with the lowest standard deviation of 0.83. Our *Composite* aggregation strategy *Multi1* has the second-lowest standard deviation at 0.89, while the average rating is 4.56, which places it in the middle. The distance measure is not enough good to evaluate the quality of recommendations, e.g. *Average+Normalize* performed best in our experiments but has averaged at 4.44, which is in the lower half comparing to the rest. The most controversial strategies are *Multiplicative*, and *Approval voting 50* – their averages are the lowest at 3.95 and 4.03, and they have the largest variance. The reason is that they often recommend only a few items. The average amount of recommended places is close to eight, in some cases, these strategies resulted in just a single or no recommendations at all. *Average without Misery* and *Approval voting 33* produced on average 13 recommendations, due to being more aggressive to less desired features or features desired by single members. The participants mentioned that they rarely see any difference between profile strategies *Type 1* and *Type 2*. *Type 3* has produced higher average distance of recommendations – 0.52, while *Type 1* and *Type 2* produced very similar average distances and standard deviations.

V. Summary

The presented web application *Pix2Trips* aims to support the decision-making process of small groups of tourists who want to decide which places to visit in a new city. We have adapted an image-based preferences elicitation method for individuals to work for groups using various aggregation strategies, including two proposed in this paper. Since this is a same-time, different-place type of collaboration, real-time interactions are supported, such as messages exchange and actions connected with the recommendations themselves. The results obtained from user evaluation are promising and suggest that this approach could be used in the real world. Multiple participants in the evaluation found the preferences specification process interesting and mentioned that they would use such a system for other cities. The majority shared that their preferences were taken into account in the group recommendations, which indicates that individual preferences could be elicited by pictures and then incorporated into a group model.

We used distance as the primary measure for determining recommendations quality, which shows how close a place is to the group model. With user ratings, we chose a weak correlation between distance and perceived quality. It is not enough to determine whether one set of parameters produces a better result than another one. According to the experiments with artificial groups, the best aggregation strategy was *Average + Normalize*, which consistently produced recommendations closest to the group model. During the user evaluation, however, it was positioned below the average rating obtained for all recommendations. Generally, different aggregation strategies exhibit different effects that could be useful in various contexts – it could be a subject of further study. We could not determine a single best strategy to use in all cases. *Average* and *Average without Misery* were often rated above the midpoint, but the second can produce fewer recommendations than expected. *Multiplicative* has exhibited interesting property – the resulting places consider only the common preferences between all members. The *Composite* strategy used in evaluation (*Multi1*) has been rated between *Least Misery* and *Most Pleasure*, but its rating has the lowest variance. The data preparation process could be improved because emotions evoked by pictures are somewhat subjective and can differ between groups. Pictures database has to be crafted by multiple people to achieve some level of generality. Also, assigning pictures to places should be done by experts who have visited the places under consideration. Another aspect is the process of assigning features to places. Doing it manually is tedious and error-prone, especially if multiple cities are to be supported. It would be better to use text mining and extract the information from places' descriptions and user reviews.

References

[1] Johansen, R.: Teams for Tomorrow (groupware). In: Proc. of the Twenty-Fourth Annual Hawaii International Conf. on System Sciences. **3**, Decision support and knowledge-based systems and collaboration technology, 521–534, IEEE Comput. Soc.Press (1991).

[2] Werthner, H., et al.: Future Research Issues in IT and Tourism. Information Technology & Tourism, **15**(1), 1–15 (2015).

[3] Felfernig, A. et al: Algorithms for Group Recommendation. In Group Recommender Systems, Springer Intern. Publishing, (2018) 27–58.

[4] Masthoff, J.: Group Modeling: Selecting a Sequence of Television Items to Suit a Group of Viewer. In Personalized digital television. Springer, 93–141 (2004).

[5] Nguyen, T. N.: Conversational Group Recommender Systems. In Proc. of the 25th Conf. on User Modeling, Adaptation and Personalization, 331–334 (2017).

[6] Neidhardt, J. et al: A Picture-Based Approach to Recommender Systems. Information Technology & Tourism **15**(1), 49–69 (2015). DOI: 10.1007s40558-014-0017-5.

[7] Goldberg, L. R.: An Alternative "Description of Personality": The Big-Five Factor Structure. Journal of Personality and Social Psychology **59**(6), 1216–1229 (1990).

[8] Gibson, H., Yiannakis, A.: Tourist Roles - Needs and the Lifecourse. Annals of Tourism Research **29**(2), 358–383 (2002).

[9] Glatzer, L., Neidhardt, J., Werthner, H.: Automated Assignment of Hotel Descriptions to Travel Behavioural Patterns. In Information and Communication Technologies in Tourism, Stangl, B. and Pesonen, J., Eds., Springer, 409–421 (2018).

[10] Berger, H. et al: Quo Vadis Homo Turisticus? Towards a Picture-Based Tourist Profiler. In Information and Communication Technologies in Tourism, Sigala, M. et al. Eds., Springer, Vienna 87–96 (2007).

[11] Linaza, M. T. et al: Image-Based Travel Recommender System for Small Tourist Destinations. In Information and Communication Technologies in Tourism , Law, R. et al Eds., Springer Vienna, 1–12 (2011).

[12] Masthoff, J.: Group Recommender Systems: Combining Individual Models. In Recommender Systems Handbook, Ricci, F. et al Eds., Springer, 677–702 (2011).

[13] Ardissono, L. et al: INTRIGUE: Personalized Recommendation of Tourist Attractions for Desktop and Hand Held Devices. Applied Artificial Intelligence **17**(8-9), 687–714 (2003). DOI:10.1080713827254

[14] Nguyen, T. N. and Ricci, F.: A Chat-Based Group Recommender System for Tourism. Information Technology & Tourism **18**(1-4), 5–28 (2018).

[15] Álvarez Márquez, J. O., Ziegler, J.: Hootle+: A Group Recommender System Supporting Preference Negotiation. In Collaboration and Technology, Yuizono, T. et al Eds., **9848**, Springer, 151–166 (2016). DOI: 10.1007978-3-319-44799-5_12.

[16] Binucci, C. et al: Designing the Content Analyzer of a Travel Recommender System. Expert Systems with Applications **87**, 199–208 (2017).

[17] Brooke, J.: SUS – A Quick and Dirty Usability Scale. In Usability evaluation in industry. CRC Press, 189–194 (1996).

[18] Bangor, A. et al: An Empirical Evaluation of the System Usability Scale. Intern. Journal of Human-Computer Interaction. **24**(6), 574–594 (2008).

[19] Pu, P. and Chen, L.: A User-Centric Evaluation Framework of Recommender Systems. In Proceedings of the ACM RecSys Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces **612**, 14–21, (2010).

[20] Hogg, M. A. and Hains, S. C.: Friendship and Group Identification: A New Look at the Role of Cohesiveness in Groupthink. European Journal of Social Psychology **28**(3), 323–341 (1998).

[21] Dawes, J.: Do Data Characteristics Change According to the Number of Scale Points Used? An Experiment Using 5-Point, 7-Point and 10-Point Scales. International Journal of Market Research **50**(1), 61–104 (2008).

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 147–150

DOI: 10.15439/2021F79
ISSN 2300-5963 ACSIS, Vol. 25

Analysis of the Effect of News Sentiment on Stock Market Prices through Event Embedding

Sashank Sridhar
College of Engineering Guindy,
Anna University, Chennai, India
Email:
sashank.ssridhar@gmail.com

Sowmya Sanagavarapu
College of Engineering Guindy,
Anna University, Chennai, India
Email:
sowmya.ssanagavarapu@gmail.com

***Abstract*—Stock market price prediction models have remained a prominent challenge for the investors owing to their volatile nature. The impact of macroeconomic events such as news headlines is studied here using a standard dataset with closing stock price rates for a chosen period by performing sentiment analysis using a Random Forest classifier. A Bi-LSTM time-series forecasting model is constructed to predict the stock prices by using the polarity of the news headlines. It is observed that Random Forest Classifiers predict the polarity of news articles with an accuracy of 84.92%.**

I. Introduction

NEWS plays a significant role in the investment world as it provides information to the investors to make decisions in the stock markets. It is capable of shaping and influencing the emotions and opinions of people, driving the decision to buy or sell in markets. Recently, an example of this was the world markets crashing in March 2020 [1] during the COVID-19 global pandemic. Due to the imposition of nationwide lockdowns forcing businesses to close down and stop their ongoing activities, investors were faced with uncertainty, leading to the markets across the world crashing in March 2020.

Analysis of media sentiments enables performing text analysis to determine its opinion and the subjectivity. In his Economic Research paper, Samuel P. Fraiberger of the World Bank [2] has shown that news sentiment acts as an important predictor of daily stock returns in stock markets.

Sentiment analysis [3], or opinion mining, is a Natural Language Processing (NLP) technique to determine the polarity of the text sentiment (positive, negative, or even neutral). Machine learning architectures such as Support Vector machines, Boosting and Bagging algorithms and Random Forests perform this analysis by assigning sentiment scores to the categories within a phrase in a sentence to determine polarity.

Random Forests is a Machine Learning Algorithm that is built using multiple decision trees merged together for an accurate and stable prediction. This algorithm also adds randomness to the data for enhancing its performance, while training using the data bagging algorithm.

Stock market price prediction models have helped in the determination of asset investments for maximizing individual profits. The complexity of the time-series forecasting tasks are handled by Bidirectional Long Short Term Memory (Bi-LSTM) [4].

In this paper, sentiment analysis is performed on the varying sets of news headlines dataset collected for each day to analyze the polarity of the data as positive or negative. The positive class refers to the headlines which led to the increase in the stock price the next day and the negative class indicated the drop in the stock price. Using a random forest classifier, the data was studied and the results were analyzed for the impact of polarity prediction on stock price forecasting using Bi-LSTMs.

The rest of the paper is organized as follows. Section II gives the summary of some of the best works in stock market price prediction modelling. The design of the system for market price prediction using news and the sentiment analysis model for prediction of rate change is given in Section III. The implementation details of the models are given in Section IV. The results obtained from the implemented price prediction system is presented in Section V. The summarization of the project and the proposed future work is given in Section VI.

II. Related Works

In this section, some of the recent works in stock market price prediction using sentiment analysis have been summarized.

Stock movement prediction model using dilated causal convolutions and transformer modelling was discussed by Daiya and Lin [5]. They extracted features from the data to feed into a multi-head self-attention model by considering financial indicators and news data. A basic reinforcement learning policy and reward function to match with their performance was used in their model. By implementing multimodal learning, the model aimed to maximize forecasted profits through asset allocation based on the prediction modelling.

Case studies dealing with the effect of public sentiment in stock market price predictions using big data analysis have also been published. Bourezk et al [6] used machine learning algorithms to analyze the relationship between the general public view regarding a stock and its evolution within the Moroccan Stock Exchange. Malawana and Rathnayaka [7] performed sentiment analysis on market related announcements in the news to extract positive, negative and neutral opinions. Using Naive Bayes and Linear Regression models for this, they performed detections of sentiment class within a Big Data distributed Environment.

III. System Design

The overall architecture for the prediction of stock prices and the rate of movement is given in Figure 1. Multitask architecture comprises a sentiment analysis module which determines the direction of movement of stocks from news headlines.

I.A. Data Set Description

The dataset used is collected from [8] and it consists of stock price data for Dow Jones Industrial Average (DJIA) and corresponding news article headlines regarding the stock index from the period between 2008-08-08 to 2016-07-01. The headlines for each day are annotated as either 0 when the close price decreased compared to the previous day and 1 when the close price stayed the same or rose compared to the previous day.

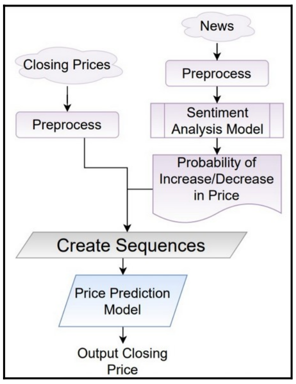


Fig. 1. Overall Multitask Architecture for Prediction

I.B. Preprocessing the Price Dataset for Price Prediction and Rate of Change Analysis

The price dataset is first normalized to ensure all the values are in the range (0,1) that helps the model to converge at the local minima at a faster rate. The dataset is converted into input sequences of length n and the corresponding output price is the (n+1)th price in the sequence. This creates a sliding window of fixed length whose output corresponds to the forecasted prices.

I.C. Preprocessing News Headlines for Sentiment Analysis

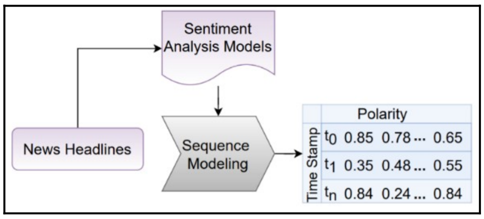
Data cleaning is performed and each word in a sentence is converted into a vector form to be modelled by assigning a Term frequency - inverse document frequency (Tf-IDF) score [9].

I.D. Multitask Learning

Multitask learning [10] aims to learn individual sub-tasks separately and use those learnings inductively to solve a main task by identifying the dependence between the tasks. Separate multitask models are built to predict the rate of change of stock prices and to predict the actual stock price itself. The subtasks involve modelling the prices and identifying the sentiment of the news headlines and these subtasks act as Level-0 models.

I.E. Event Embedding with Sentiment Analysis

The vectorized headlines are fed to different machine learning models in order to predict their positive sentiment corresponding to the close prices being steady or increasing and negative sentiment corresponding to the close prices decreasing. The obtained probabilities of sentiment from the machine learning models act as events [11] that contribute to the price prediction model. The events are converted into sliding windows to correspond to the sliding window of prices and are given as input to the overall price prediction multitask model. Figure 2 shows how event sequences are generated using the polarity derived from the sentiment analysis models.



	Polarity
t_0	0.85 0.78 ... 0.65
t_1	0.35 0.48 ... 0.55
t_n	0.84 0.24 ... 0.84

Fig. 2. Event Embedding of Sentiment Polarity

I.F. Price Modelling

Sequences of stock prices are proposed to be modelled using Long Short-Term Memory (LSTM) models and the modified Bi-Directional LSTM (Bi-LSTM) models. LSTM models make use of gates to determine if the data at a particular node should be retained or not from the cell state to map future and past interrelations between the data. Bi-LSTMs use the LSTM cells to map the dependencies between sequences in both the forward and reverse directions.

I.G. Layer-1 Meta Classifier

Once the subtask models are built the output of the models are fed to a Layer-1 Meta-Classifier that relates both the tasks. The meta-classifier used is an Artificial Neural Network (ANN). The outputs of the price prediction sub-model and the sentiment analysis model are given as inputs to the ANN which then predicts the output closing price of the given stock index.

IV. System Implementation

The implementation details of the system are given in this section.

A. Dataset Split

1) Sentiment Analysis Model

The sentiment analysis model has top 25 news headlines for each day from 2008-08-08 to 2016-07-01. The dataset was divided in a train-test ratio of 80: 20 as seen in Table I.

2) Price Prediction Model

The dataset was divided into a train-test ratio of 80:20 with the training set consisting of prices for 1863 days and the testing set consisting of prices for 378 days. Prices are transformed to form a sliding window with an input sequence length of 50 and an output sequence length of 1.

B. Sentiment Analysis

The Tf-IDF score is calculated for news headlines using sklearn's Tf-IDF Vectorizer. Each day's headlines are annotated with labels of 0 or 1 corresponding to a decrease or increase in prices on that day. Different machine learning algorithms are implemented with their corresponding parameters as seen in Table II.

C. Price Prediction

The input closing prices is first normalized using sklearn's MinMaxScaler which ensures that all prices remain between 0 and 1. The model comprises 3 Level-0 models corresponding to modelling of prices, negative sentiment and positive sentiment probabilities of the news headlines. The sub-models that learn the positive and negative sentiment of the headlines. The outputs of all the three sub-models are concatenated and fed to a Level-1 meta-classifier with ReLu Activation. The prediction model is trained for 1000 epochs in batches of size 8 with RMSProp as the optimizer and MSE as the loss function

V. Results and Analysis

The results obtained from the stock price prediction model and sentiment analysis models are presented in this section along with their analysis.

A. Evaluation metrics for the sentiment analysis model using Machine Learning algorithms

The standard dataset contains day-by-day news headlines along with whether the stock market price increased (positive class) or decreased/remained the same (negative class) the next day for sentiment analysis classification of the text as positive and negative class. It is observed from Table III, that the Random Forest (RF) algorithm has outperformed the other algorithms with the no. of headlines at 25. This RF model [12] is composed of a number of decision tree classifiers that help to identify the important gestures from the dataset for its high performance.

TABLE II.
DATASET SPLIT FOR SENTIMENT ANALYSIS MODEL

Data Type	Train Set	Test Set
Number of Days	1863	378
Number of News Headlines	50301	10206
Headlines with Positive Sentiment	26865	5184
Headlines with Negative Sentiment	23436	5022

TABLE II.
PARAMETERS OF SENTIMENT ANALYSIS CLASSIFIERS

Parameter	Value
Random Forest	n_estimators=200, criterion='entropy'
Support Vector Machine	kernel='linear'
Adaboost Classifier	n_estimators=200
Bagging Classifier	base_estimator=Support Vector Classifier, n_estimators=10, random_state=0
Decision Tree Classifier	random_state=0

B. Evaluation metrics for the Random Forest sentiment analysis model

Sentiment analysis was carried out by using RF classifier on a day's news headlines to predict the increase or decrease in the stock price for the next day. The random forest-based machine learning algorithm was tested with multiple numbers of headlines chosen on each run to identify and record its optimal performance. The positive class or class-1 refers to the news headlines that predicted the increase in stock price the next day and negative class or class-0 when the closing price of the stock remained the same or decreased. From Table IV, it's observed that it performed best with 25 headlines, reaching a performance accuracy of 84.92. The confusion matrix for the RF based sentiment analysis model is given in Figure 3.

C. Calculation of sentiment score for positive and negative sentiment

The sentiment score is calculated for every news headline from the dataset. In a headline with a positive sentiment, count for each word in both the positive counter and the total words in the dataset counter; likewise, for each word in a negative sentiment headline, count for that word in both the negative counter and the total words counter is increased.

The most commonly occurring words belonging to the news headlines from the positive and negative sentiment are extracted and visualized in Figure 4. The headlines that referred to new releases seemed to have one of the highest impacts on the stock market prices. News related to hacking, sanctions and scandals have had the highest negative impact resulting in the fall of the stock prices.

D. Prediction of Stock Prices with News Sentiment Analysis

The daily stock prices are plotted in the graph in Figure 5 and compared with the predicted values from the Bi-LSTM model trained with news headlines for performance analysis. It is observed that the prediction results with top 25 news headlines shows that the model is able to predict the positive

TABLE III.
PARAMETERS OF SENTIMENT ANALYSIS CLASSIFIERS

Model	No. of headlines					
	1	5	10	15	20	25
RF	84.1	83.6	84.7	83.6	81.7	84.9
SVM	82.8	83.9	83.3	83.6	84.6	84.6
Adaboost	63.7	72.2	71.9	74.6	73.5	74.3
Bagging	80.6	81.4	81.4	81.4	81.4	81.9
Decision Tree	82.0	81.7	77.5	81.5	77.5	80.68

or negative trend change on the next day with a high value of accuracy through NLP techniques.

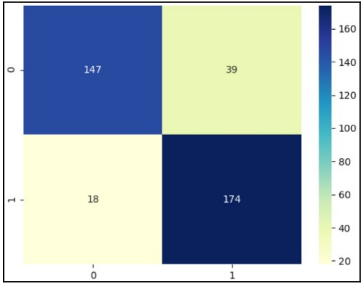


Fig. 3. Confusion Matrix obtained from the Random Forest Model Sentiment Analysis Classification

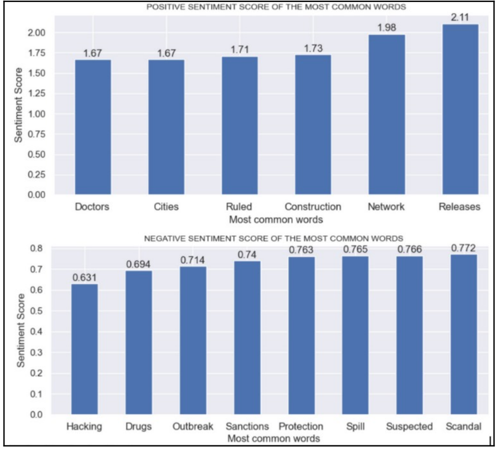


Fig. 4. Most Common Words from the Dataset with a) High Positive Score and b) High Negative Score

IV. Conclusion

The work explores the effect of public sentiment through news headlines on stock market prices. To accomplish that, a Random Forest classifier-based sentiment analysis model is constructed using day-by-day news headlines dataset along with the variation of Close Stock price. The sentiment analysis model identified the headlines associated with positive and negative sentiment for further analysis. The constructed Multitask Bi-LSTM based stock price prediction model was used for predicting the close price rates with news headlines dataset. A deep neural network architecture-based stock price prediction model is to be constructed to experiment with using trained weights from the sentiment analysis performed for optimizing the learning weights of the model further with attention-based deep neural architectures. This model would be analyzed to calculate the rate of change of price with chosen top-n headlines for positive and negative sentiment in the news articles.

Table IV.
Performance of the Random Forest Sentiment Analysis Model

Evaluation Metric	**Performance of the model in %**
Accuracy	84.92
Precision	85.0
Recall	85.0
F1-Score	85.0

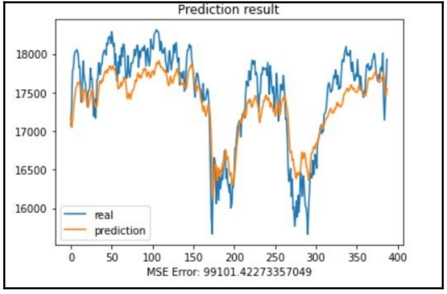


Fig. 5. Real and Prediction Value comparison for the trained Bi-LSTM Multitask Model with News Headlines Data

References

1. J.-J. Ohana, S. Ohana, E. Benhamou, D. Saltiel, and B. Guez, “Explainable AI Models of Stock Crashes: A Machine-Learning Explanation of the Covid March 2020 Equity Meltdown,” SSRN Electronic Journal, 2021, doi: http://dx.doi.org/10.2139/ssrn.3809308.
2. S. P. Fraiberger, D. Lee, D. Puy, and R. Rancière, “Media Sentiment and International Asset Prices,” NBER Working Papers 25353, National Bureau of Economic Research, Inc., 2018.
3. M. Skuza and A. Romanowski, “Sentiment Analysis of Twitter Data within Big Data Distributed Environment for Stock Prediction,” in 2015 Federated Conference on Computer Science and Information Systems (FedCSIS), pp. 1349–1354, 2015, doi: 10.15439/2015F230.
4. D. Ruta, L. Cen and Q. H. Vu, "Deep Bi-Directional LSTM Networks for Device Workload Forecasting," in 2020 15th Conference on Computer Science and Information Systems (FedCSIS), pp. 115-118, 2020, doi: 10.15439/2020F213.
5. D. Daiya and C. Lin, “Stock Movement Prediction and Portfolio Management via Multimodal Learning with Transformer,” in ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3305–3309, 2021, doi: 10.1109/ICASSP39728.2021.9414893.
6. H. Bourezk, A. Raji, N. Acha, and H. Barka, “Analyzing Moroccan Stock Market using Machine Learning and Sentiment Analysis,” in 2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), pp. 1–5, 2020, doi: 10.1109/IRASET48871.2020.9092304.
7. M. V. D. H. P. Malawana and R. M. K. T. Rathnayaka, “The Public Sentiment analysis within Big data Distributed system for Stock market prediction– A case study on Colombo Stock Exchange,” in 2020 5th International Conference on Information Technology Research (ICITR), pp. 1–6, 2020, doi: 10.1109/ICITR51448.2020.9310871.
8. J. Sun, “Daily News for Stock Market Prediction, Version 1,” kaggle.com, 2016. https://www.kaggle.com/aaron7sun/stocknews (accessed May 23, 2021).
9. J. A. Reyes-Ortiz, M. Bravo, and H. Pablo, “Web Services Ontology Population through Text Classification,” in 2016 Federated Conference on Computer Science and Information Systems, pp. 491–495, 2016, doi: 10.15439/2016F332.
10. A. Gillioz, J. Casas, E. Mugellini and O. A. Khaled, "Overview of the Transformer-based Models for NLP Tasks," in 2020 15th Conference on Computer Science and Information Systems (FedCSIS), pp. 179-183, 2020, doi: 10.15439/2020F20.
11. P. Maciąg, “Efficient Discovery of Top-K Sequential Patterns in Event-Based Spatio-Temporal Data,” in 2018 Federated Conference on Computer Science and Information Systems, pp. 47–56, 2018, doi: 10.15439/2018F19.
12. J. Lindén, S. Forsström, and T. Zhang, “Evaluating Combinations of Classification Algorithms and Paragraph Vectors for News Article Classification,” in 2018 Federated Conference on Computer Science and Information Systems, pp. 489–495, 2018, doi: 10.15439/2018F110.

14th International Workshop on Computational Optimization

MANY real world problems arising in engineering, economics, medicine and other domains can be formulated as optimization tasks. These problems are frequently characterized by non-convex, non-differentiable, discontinuous, noisy or dynamic objective functions and constraints which ask for adequate computational methods.

The aim of this workshop is to stimulate the communication between researchers working on different fields of optimization and practitioners who need reliable and efficient computational optimization methods.

Topics

The list of topics includes, but is not limited to:

- combinatorial and continuous global optimization
- unconstrained and constrained optimization
- multiobjective and robust optimization
- optimization in dynamic and/or noisy environments
- optimization on graphs
- large-scale optimization, in parallel and distributed computational environments
- meta-heuristics for optimization, nature-inspired approaches and any other derivative-free methods
- exact/heuristic hybrid methods, involving natural computing techniques and other global and local optimization methods
- numerical and heuristic methods for modeling

The applications of interest are included in the list below, but are not limited to:

- classical operational research problems (knapsack, traveling salesman, etc)
- computational biology and distance geometry
- data mining and knowledge discovery
- human motion simulations; crowd simulations
- industrial applications
- optimization in statistics, econometrics, finance, physics, chemistry, biology, medicine, and engineering
- environment modeling and optimization

Best Paper Award

The best WCO'21 paper will be awarded during the social dinner of FedCSIS 2021.

The best paper will be selected by WCO'21 co-Chairs by taking into consideration the scores suggested by the reviewers, as well as the quality of the given oral presentation.

Technical Session Chairs

- **Fidanova, Stefka,** Bulgarian Academy of Sciences, Bulgaria
- **Mucherino, Antonio,** INRIA, France
- **Zaharie, Daniela,** West University of Timisoara, Romania

Program Committee

- **Abud, Germano,** Universidade Federal de Uberlândia, Brazil
- **Bonates, Tibérius,** Universidade Federal do Ceará;, Brazil
- **Breaban, Mihaela,** West University of Timisora, Romania
- **Gruber, Aritanan,** Federal University of ABC, Santo André, Brazil
- **Hosobe, Hiroshi,** Hosei University, Japan
- **Kouichi, Hirata,** Kyushu Institute of Technology, Kawazu, Japan
- **Lavor, Carlile,** IMECC-UNICAMP, Brazil
- **Micota, Flavia,** West University of Timisora, Romania
- **Muscalagiu, Ionel,** Politehnica University Timisoara, Romania
- **Stoean, Catalin,** University of Craiova, Romania
- **Tami, Tamir,** School of Computer Science, The Interdisciplinary Center, Herzliya, Israel
- **Wang, Yifei,** Georgia Institute of Technology, USA
- **Zilinskas, Antanas,** Vilnius University, Lithuania

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 153–162
DOI: 10.15439/2021F18
ISSN 2300-5963 ACSIS, Vol. 25

Algorithms for the Safe Management of Autonomous Vehicles

Mourad Baïou
LIMOS Lab. CNRS/UCA,
Clermont-Ferrand, France

Aurelien Mombelli, Alain Quilliot
Labex IMOBS3, LIMOS Lab.
CNRS/UCA, Clermont-Ferrand, France

Lounis Adouane
HEUDYASIC Lab.CNRS and
UTC, Compiègne, France

Zhengze Zhu
LIMOS and InstitutPascal Labs
UCA/CNRS, Clermont-Ferrand, France

***Abstract*—We deal here with a fleet of autonomous vehicles which is required to perform internal logistics tasks inside some protected area. This fleet is supposed to be ruled by a hierarchical supervision architecture, which, at the top level distributes and schedules *Pick up and Delivery* tasks, and, at the lowest level, ensures safety at the crossroads and controls the trajectories. We focus here on the top level, while introducing a time dependent estimation of the risk induced by the traversal of any arc at a given time. We set a model, state some structural results, and design, in order to route and schedule the vehicles according to a well-fitted compromise between speed and risk, a bi-level algorithm and a A* algorithm which both relies on a reinforcement learning scheme.**

I. Introduction

INTELLIGENT vehicles, provided with an ability to move with some level of autonomy, are the new hot spot in Mobility [1]. Still, determining what can be exactly done with new generations of autonomous or semi-autonomous vehicles able to follow their own way without being physically tied to any kind of track (cable, rail, …) remains an issue. Most people are doubtful about the prospect of seeing such vehicles moving without any external control inside crowded urban areas.

Fig. 1 An Autonomous Vehicle

Instead they foresee that the use of those vehicles is likely to be restricted to protected areas and professional purposes: moving free access vehicles inside large parking areas, performing rural or urban logistics or replacing too constrained AGV (*Autonomous Guided Vehicles*) inside warehouses or industrial structures (see [1]).

This point of view raises the general challenge of monitoring a fleet of such a vehicles, required to perform internal logistics tasks while safely interacting with workers, machines and standard vehicles. Related decisional problems are at the intersection of Robotics and Operations Research.

When it comes to the management of such systems, current trend is to the implementation of a hierarchical supervision architecture, relying on 2 or 3 levels:

- The first level, or *embedded* level, is defined by the monitoring and sensing devices which are embedded inside the vehicles, compute the trajectories in real time and adapt them to the possible presence of obstacles: currently, most effort from the robotics community remains devoted to this embedded level (see [9, 12, 13, 17]), which mostly involves optimal control and artificial perception techniques;
- The second one, or *midle* one, is in charge of the supervision of small *tricky* areas, like for instance crossroads or loading/unloading spots (see Figure 2).

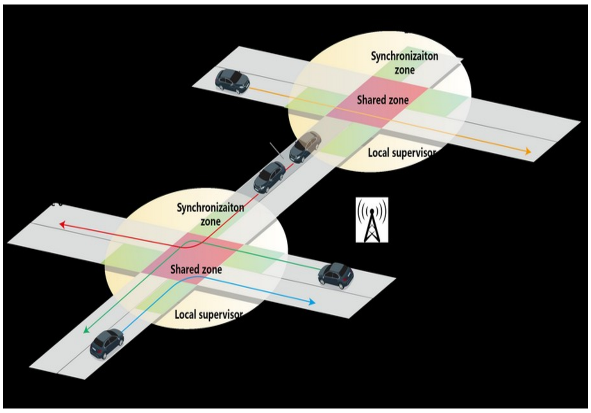


Fig. 2 A Hierarchical Supervision Architecture

Working as a mediator agent, it sends signals and instructions to the vehicles in order to regulate their transit and to avoid them to collide when they get through those areas. This level have been motivating a

This work was supported by ANR, Labex IMOBS3

rise in interest for the last years (see [5, 11, 15]), and sometimes a confusion with the *embedded* level: in many cases, hypothesis is set that all vehicles involved are run by the same embedded software and exchange perfect information; this become equivalent to supposing the existence of a local external mediator.

- The third one, or *global* one, consists in tactical dynamic planning and routing of the fleet, and the distribution of *Pick up and Delivery* (PDP) tasks among the vehicles (see [4, 7, 12, 21]).

Depending on the complexity and the size of the system, the second level may merge with either the first one or the last one. In any case, a true challenge is about the synchronization of those monitoring levels, which correspond to distinct time scales and purposes, and the design of communication protocols which will allow them to interact.

Our goal here is to deal with the *Global Monitoring* level. By some aspects, related problems may be viewed as cases of well-known *Pick up and Delivery* problem (see [4]), since in most cases a task will consist for a vehicle in moving to some place, performing some loading or unloading transaction and keep on. But two specific features are going to bring its specificity to this PDP variant:

- The time horizon of autonomous or semi-autonomous vehicles is usually somewhat short: decisions have to be taken fast, in a dynamic context, and decisional processes must take into account the communication infrastructure [20] and the way the global supervisor can be provided, at any time, with a representation of the current state of the system and its short term evolution;
- As soon as autonomous or semi-autonomous vehicles are involved, safety is at stake (see [2, 16, 17, 18]). The global supervisor must compute and schedule routes in such a way that not only tasks are going to be performed fast (standard industrial efficiency) but also that local and embedded supervisors will perform their job more easily. In other words, risk minimization should be a criterion for a good schedule.

A consequence is that performing the top level supervision of a fleet of autonomous vehicles requires disposing at any time of an accurate representation of the current state of the system and its short term evolution. This representation should enable us able to quantify the risk induced by an additional vehicle, which enters into the transit network and is asked to follow a given trajectory. We are not going to directly address this issue, which is complex (see [18, 23]). Instead, we are going to suppose that, at the time when we are trying to schedule this vehicle, we are provided with a procedure which, to any arc (x, y) of the transit network and any time value t, computes an estimation of the risk resulting from the presence of our vehicle on arc e at time t. Then our goal becomes to compute and schedule the route Γ of our vehicle, in such a way that its riding time is minimized and that induced risk estimation does not exceed some threshold *Risk_Max*. For the sake of simplicity, we shall limit ourselves to a one task tour, which means that Γ will be constrained by its starting point o and its destination point d. Described this way, our problem might be view as the search for a *constrained shortest path* [7]. But the fact that both risk and arc traversal times are time dependent makes the problem significantly more difficult (see [7]). By the same way, the *on line* feature of a system such that an autonomous vehicle fleet keeps us from relying on heavy mathematic machinery like MILP models or time expanded networks [9], and impose us to design ad hoc fast heuristics.

According to this purpose, we propose here 2 algorithms: The first one is a bi-level heuristic one, whose structure may be compared to the structure of *Split* algorithms which are used in *Route First Cluster Second* algorithms for *Vehicle Routing* problems. This algorithm iteratively acts on the topology of the route Γ and next schedule the vehicle along this route according to a filtered dynamic programming procedure. The second one is a A* tree search algorithm [14], which explores a *risk expanded network*.

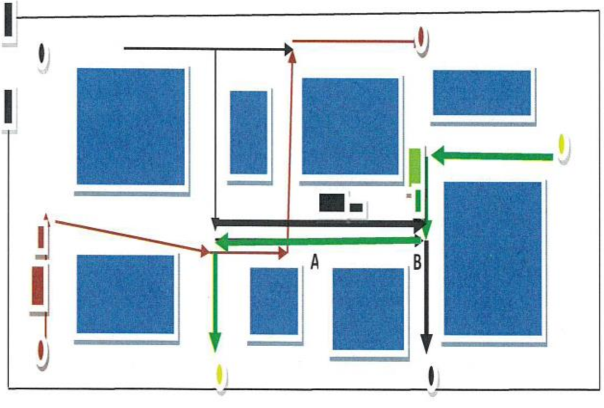
Both algorithms are designed in order to induce small computing costs and both perform a kind of auto-adaptative reinforcement learning scheme [3, 10, 13, 22], which aim at estimating a good conversion ratio *time versus risk*.

So the paper is organized as follows: in Section II we formally describe our model and state some structural results. In Section III we describe the bilevel local search heuristic [19], while in Section IV we describe the A* like algorithm. Section V is devoted to numerical experiments.

II. The model

A. Transit Network and Risk Function

Transit Network and Risk Function. We suppose that our fleet of vehicles moves inside a simple planar transit network $G = (N, A)$, N denoting the nodes of G, and A its arcs, likely to represent for instance a warehouse (see figure 3 below).



At time t = 10, both green and black vehicles are scheduled to be involved in arc (A, B) => risky area

Fig. 3. A Warehouse like Transit Network.

Every arc $e = (x, y)$ is provided with a maximal speed V_Max_e and a length D_e. We denote by *TIME* the shortest path distance induced by shortest traversal time values D_e/V_Max_e. At the time $t = 0$ when the global supervisor of the fleet needs to take a decision about target vehicle *VEH*, he knows about routes followed by the other vehicles and the tasks they are going to perform. So he is provided, for any arc $e = (x, y)$ and any future instant $t > 0$, with an estimation of the number of vehicles and obstacles which are going to be located in e at time t. This allows him to derive a risk estimation $\Pi^e(t)$ whose meaning is:

- For any small value dt, $\Pi^e(t).dt$ is the *Expected Damage* between time t and time $t+dt$ in case *VEH* moves at maximal speed V_Max_e along e during this period.

Since we practically derive [18] any function Π^e from a finite (small) set of possible activity configurations related to arc e, we suppose that this function, which translates those configurations into risk, is piecewise linear (see figure 4). We call *break points* of $\Pi^e(t)$ the values t when the value of $\Pi^e(t)$ changes.

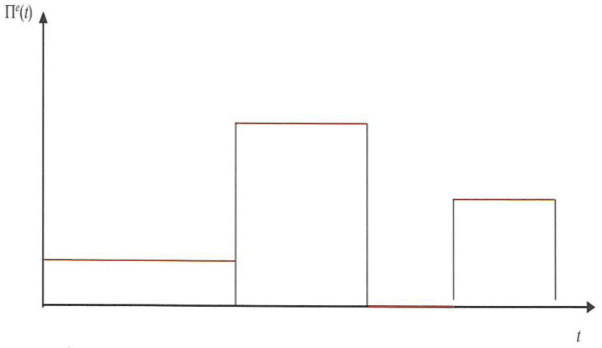


Fig. 4. A Piecewise Function Π.

Then we assume that, if *VEH* traverses arc e during some interval $[t, t+ dt]$ at speed $v \leq V\_Max_e$, then related *Expected Damage* is given by a formula:

- $Risk^e(v, t) = \Phi(v/V\_Max_e).\Pi^e(t).dt$, where Φ is a convex increasing function with values in [0, 1] and such that for any value u, $\Phi(u)$ is significantly smaller than u. Those condition are imposed in order to confirm the intuition which tells that that the slower the vehicle moves, the smaller is resulting risk. In our experiments, we shall use function $\Phi(u) = u^2$.

It comes that if vehicle *VEH* moves across arc e between time T and time $T + \delta$, according to speed function $t \text{ -> } v(t)$, then related *Expected Damage* is:

$$\int_{[T, T+\delta]} \Phi(v(t)/V\_Max_e).\Pi^e(t)\, dt.$$

B. Routing Strategies and SPRC Model

Let us suppose now that origin o and destination d are given, as nodes of the transit network $G = (N, A)$. A *routing strategy* for our vehicle, is going to be a pair (Γ, v), where Γ is a path in the network G, and v is a *speed function*, which, to any time value $t \geq 0$, makes corresponds the speed $v(t)$ of the vehicle. Clearly, if at time t, *VEH* in located on arc $e \in \Gamma$, then $v(t)$ must not exceed V_Max_e.

Path Γ may be viewed in a standard way as a sequence $e_1, \ldots, e_n$ of arcs of G. If we set $T(0)= 0$ and denote by $T(i)$ the time when *VEH* arrives to the end-node of e_i, then values $T(i)$ are completely determined speed function $t \text{ -> } v(t)$. Then we set:

- $G\_Time(\Gamma, v) = T(n) =$ *global duration* of the routing strategy (Γ, v);
- $G\_Risk(\Gamma, v) = \Sigma_i \int_{[T(i-1), T(i)]} \Phi(v(t)/ V\_Max_e).\Pi^e(t)\, dt$ = *global risk* of the routing strategy (Γ, v).

The SPRC: ***Shortest Path Under Risk Constraint* Model**: Then our purpose becomes in a natural way to make vehicle *VEH* move from o to d while achieving small $G\_Time(\Gamma, v)$ and $G\_Risk(\Gamma, v)$ values. This looks a kind of bi-objective formulation. As a matter of fact, risk and time play very different roles inside a real industrial system, and so the risk is usually managed as a constraint: some threshold *Risk_Max* is given and the trajectory (Γ, v) of vehicle *VEH* is required to be such that resulting risk $G\_Risk(\Gamma, v)$ does not exceed threshold *Risk_Max*. It comes that our SPRC: *Shortest Path Under Risk Constraint* model comes in a natural way as follows:

> **SPRC: *Shortest Path Under Risk Constraint***:{Given the threshold *Risk_Max*, compute a routing strategy (Γ, v) such that $G\_Risk(\Gamma, v) \leq Risk\_Max$ and $G\_Time(\Gamma, v)$ is the smallest possible}

C. Structural Results

The time dependence of the transit network together with the proximity of the SPRC model with *Shortest Path Constraint* models suggests that SPRC is a complex problem. As a matter of fact, we may state:

Proposition 1: *SPRC is NP-Hard. Even if Γ is fixed, computing speed function t -> v(t) is also NP-Hard.*

Sketch of the proof: SPRC can be reduced to the *Constrained Shortest Path* problem [6]. If we fix Γ, then we can reduce resulting problem to Knapsack. □

Still, as we shall see now, SPRC may be simplified. Let us suppose that we are provided with an optimal routing strategy (Γ, v). One easily checks that:

Proposition 2: *If VEH is running along some arc e during time T and time T + δ, and if $\Pi^e(t)$ is constant between T and T + δ, then we may do in such a way that optimal speed v(t) is constant on]T, T + δ[.*

It comes that we may impose function v to be piecewise constant, with *break points* which follow the arc e of Γ and the *break points* of functions $t \text{ -> } \Pi^e(t)$.

Also, we may notice that in general, (Γ, v) will achieve exactly the risk threshold *Risk_Max*:

Proposition 3: *If it happens, at some time t, that VEH is running inside an arc e in such a way that* $v(t) \neq V\_Max_e$, *then G_Risk(Γ, v) = Risk_Max.*

Sketch of Proof: We suppose the converse, and check that it is possible to make *G_Risk(Γ, v)* decrease by augmenting the speed, and so the resulting risk, on some arc e of Γ. □

As it is the case in multi-objective optimization, a natural question arises now about a possible conversion of risk into time, which could allow us to deal with a mono-objective problem. When talking about risk into time conversion, we mean a coefficient α which would tell us that adding dr to *G_Risk*(Γ, v) would be equivalent to adding α.dt to *G_Risk*(Γ, v). If it were existing, coefficient α would be a risk per time coefficient, that means a *risk speed*. Such a conversion is not possible in the general case (else our problem would be almost time-polynomial). Still, it is possible in a *local* way, that means inside any given arc $e \in \Gamma$ such that $v(t) \neq V\_Max_e$, and also in the *stationary* case when every function $t \to \Pi^e(t)$ is constant. More precisely, if, at some time t, we are located inside an arc e, then we define what we call the *Risk Speed* $rt^e(t)$ (*rt* as risk per time) of our routing strategy (Γ, v):

- $rt^e(t) = \Phi(v(t)/V\_Max_e).\Pi^e(t)$.

Then we may state:

Proposition 4: *If, at some time t, VEH is running inside an arc e at speed* $v(t) \neq V\_Max_e$, *and if t is not a break point for piecewise function* Π^e *then the quantity* $rt^e(t)$ *is independent on t (but not on e). Besides, in the specific case when functions* Π^e *are constant for any arc e in Γ, then* $rt^e(t)$ *is independent on t and e, as soon as constant speed* $v_e = v(t)$ *is different from* V_Max_e.

Sketch of the Proof: It is a matter of applying Kuhn-Tucker local optimality conditions for constrained optimization, to the gradient vectors of quantities *G_Risk*(Γ, v) and *G_Time*(Γ, v). □

Remark 1: Above value $rt^e(t)$, computed for t such that is *VEH* is located on e at time t with $v(t) \neq V\_Max_e$, is independent on t but dependent on e, as we may see through the following example (Figure 5):

- Path Γ contains 2 arcs, e_1 and e_2, both with length 1 and maximal speed 2. Function Π^{e2} is constant and equal to 1. Function Π^{e1} takes value 2 for $0 \leq t \leq 1$, and a very large value M (for instance 100) for $t > 1$ (see figure 5). *Risk_Max* = 3/4; Function Φ is: $u \to \Phi(u) = u^2$.

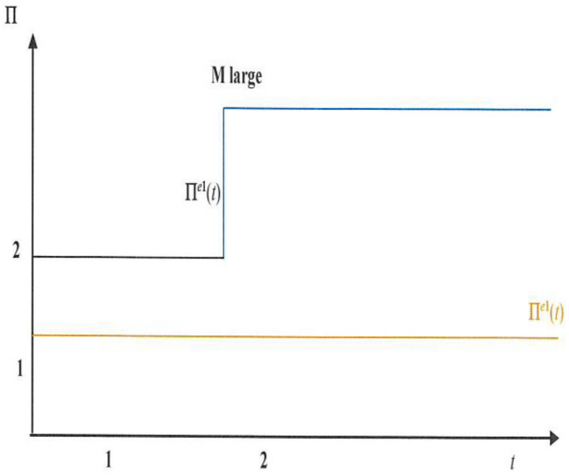


Fig. 5. Functions Π^{e1} and Π^{e2}.

- Then we see that *VEH* must go *fast* all along the arc e_1, in order to get out of e_1 before this arc becomes very risky. That means that its speed is equal to 1 on e_1, and that its *risk speed* is equal to ½. Next it puts the brake, in the sense that its speed remains equal to 1 but its *risk speed* decreases to ¼. It is easy to check that this routing strategy is the best one, with *G_Risk*(Γ, v) = ¾ and *G_Time*(Γ, v) = 2.

D. Risk Driven Reformulation of the SPRC Model

Above results allow us to significantly simplify our SPRC model by replacing the search for speed $t \to v(t)$, likely to be volatile, by the search for *risk speed* $e \to rt^*_e$, where is the *risk speed* value which corresponds to arc e in Γ according to the first part of Proposition 4. It comes that we define a *risk driven routing strategy* as a pair (Γ, rt^*) where:

- Γ is a path, that means a sequence $\{e_1,\dots, e_n\}$of arcs, which connects origin node o do destination node d;
- rt^* associates, with any arc e *in* Γ, related *risk speed* value rt^*_e which is the unique value $\Phi(v(t) V\_Max_e).\Pi^e(t)$ for any t such that *VEH* is located inside arc e_i and $v(t) \neq V\_Max_e$. Notice that if $v(t) = V\_Max_e$ then $rt(t) = \Pi^e(t)$.

Reconstructing a routing strategy (Γ, *v*) from a *risk driven routing strategy* (Γ, *rt).** Let us suppose that we know value rt^*_e related to arc e of Γ. Then, at any time t when *VEH* is inside arc e, and which is not a *break point* for function Π^e, we have: (E1)

- $v(t) = V\_Max_e$ and $dR(t)/dt = rt(t) = \Pi^e(t)$, where $R(t)$ denotes the cumulative risk between 0 and t.

or

- $v(t) = V\_Max_e$ and $rt^*_e = \Phi(v(t)/ V\_Max_e).\Pi^e(t)$ and $dR(t)/dt = rt(t) = rt^*_{e.}$.

Speed $v(t)$ is obtained by solving the equation $rt^*_e = \Phi(v(t)/V\_Max_e).\Pi^e(t)$ and next comparing it with V_Max. Since both $v(t)$ and $rt(t)$ are piecewise constant on e, we see that we may scan the arc sequence $\{e_1,\dots, e_n\}$and get, through a simple iterative process, both time $T(i)$ when *VEH* arrives at the end-node of e_i and related cumulative risk $R(T(i))$. That means that the knowledge of (Γ, rt^*) allows us to reconstruct standard routing strategy (Γ, v^*).

According to this and proposition 4, SPRC may be rewritten as follows (we extend previous notations *G_Time*(Γ, v^*) and *G_Risk*(Γ, v^*) by denoting by *G_Time*(Γ, rt^*) and *G_Risk*(Γ, rt^*) respectively the time value and risk value of a *risk driven routing strategy* (Γ, rt^*):

> **SPRC *Risk Driven* Reformulation**: {Compute *risk driven routing strategy* (Γ, rt^*) such that *G_Risk*(Γ, rt^*) $\leq$ *Risk_Max* and *G_Time*(Γ, rt^*) is the smallest possible}.

III. A First Bilevel Algorithm

We discuss here a bi-level heuristic algorithm [19] whose main iterative loop works in 2 steps:

BL_RCSP Algorithm.

```
Intialize some path $\Gamma$ from $o$ to $d$; Not Stop;
While Not Stop do
        1st step: Schedule $\Gamma$;     (*Low level step*)
        2nd step: Improve $\Gamma$;      (*Top level step*)
                  If Fail(Improve) then Stop;
Keep the best solution $\Gamma$ ever obtained.
```

The *Schedule* step considers path Γ as being fixed, and deals with the problem of computing values rt^*_e, e in Γ. Let us recall (Proposition 1) that this problem is NP-Hard. As a matter of fact, this *Schedule* step will contain the most important features of the ***BL_RCSP*** algorithm, namely those related to reinforcement learning. We shall describe it in section III.2. Meanwhile, we are going to briefly describe the *Improve* step, designed in order to modify Γ and improve its quality, and which works in a more classical way.

A. Top Level Improve Step

We suppose that some proximity threshold *S_Prox* has been fixed, and that for any two nodes x, y of the transit network G such that *TIME*(x, y) $\leq$ *S_Prox*, we are provided with a collection *Path*(x, y) of elementary path $\Omega^j{}_, j \in J(x, y)$ from x to y. Construction of collections *Path*(x, y), x, $y \in X$ such that *TIME*(x, y) $\leq$ *S_Prox*, may have been previously achieved though some preprocess. This allows us to introduce the following local transformation operator *Detour*(Γ, x, y, j), which acts on any path Γ through parameters x, y and j: x and y are 2 nodes of Γ, such that *TIME*(x, y) $\leq$ *S_Prox* and x is located before y on Γ; j belongs to $J(x, y)$.

- Then *Detour* replace the restriction $\Gamma_{x,y}$ of Γ from x to y by path $\Omega^j \in$ *Path*(x, y).

Performing the pre-process which perform the constructions of path collections *Path*(x, y), x, $y \in X$ such that *TIME*(x, y) $\leq$ *S_Prox*, allows us not to take care about path search when trying to modify Γ, and so speed the *Improve* step in a significant way.

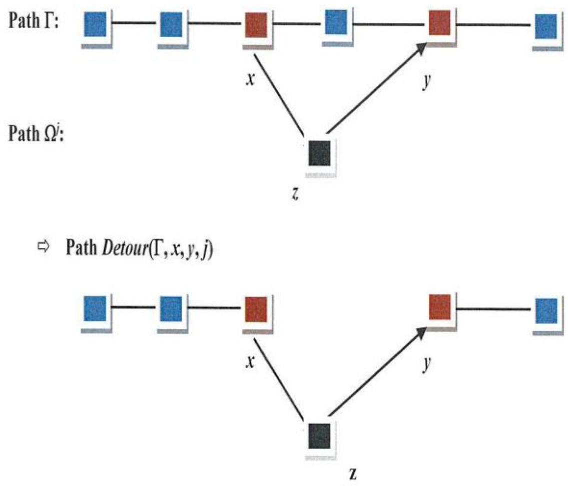


Fig. 6. *Detour* Operator

Improve step is going to drive operator *Detour* according to some standard *descent* process:

***Improve* Step**:

```
Not Stop1;
While Not Stop1 do
    Generate 3-uple $(x, y, j)$,
    Schedule Detour($\Gamma$, $x$, $y$, $j$);
    If Detour($\Gamma$, $x$, $y$, $j$) is better than $\Gamma$ then
        Stop1;
        Replace $\Gamma$ by Detour($\Gamma$, $x$, $y$, $j$)
    Else Update Stop1;
```

What remains to be told is the way we generate parameters (x, y, j) for the operator *Detour*. Once Γ has been scheduled, we are provided, for any node x in the support $X(\Gamma)$ of Γ, with:

- The time T_x, when *VEH* arrives in x, together with related cumulated risk value R_x;

Then, above *Generate* instruction focuses on 3-uples (x, y, j) such that:

- Ratio $(R_y - R_x)/(T_y - T_x)$ is large with respect to *G_Risk*(Γ, rt^*)/*G_Time*(Γ, rt^*), which suggests that sub-path $\Gamma_{x,y}$ is somewhat *crowded*;
- Path Ω^j is not very *crowed* between time T_x and time T_y, or in other words, the sum, for the arcs e of $\Omega^j{}_,$ of mean $\Pi^e(t)$ value between time T_x and time T_y is significantly smaller than the same quantity for sub-path $\Gamma_{x,y}$.

B. Low Level Schedule Step

As told before, it means the key point inside our algorithm. Basically, it consists in a dynamic programming procedure ***DP_Schedule*** whose main features come as follows:

- We denote by $e_1,\ldots, e_n$ the arcs of current path Γ, and by $x_0, \ldots, x_n$ related nodes of the node support *X*(Γ) of Γ.
- So the *time space* of ***DP_Schedule*** comes in a natural way as the set {0, 1,…, *n*} and a *state* (or *label*) at *i* is a pair (*T*, *R*), where *T* means the time when *VEH* arrives in x_i, and *R* the cumulated risk at this time. For any *i*, we shall denote as *State*(*i*) the set of *states* computed in relation with *i*. Those states will be used in order to move along arc e_i. Clearly, initial state is (0, 0) and final state (*G_Time*(Γ, *v**), *G_Risk*(Γ, *v**)) is going to be the pair (*T, R*) in *State*(*n*) with smallest *T* value and such that *R* ≤ *Risk_Max*.
- Then a *decision* at *i* comes as a value rt^*_e with arc $e = e_{i+1}$. We denote by *DEC*(*i*) he set of decisions which are tried at *i*. Resulting transition derives from equation (E1), which allows us to compute time value *T*1 and risk value *R*1 when we arrive in x_{i+1}. Clearly, decision rt^*_e will be feasible only if *R*1 ≤ *Risk_Max*.
- According to this, applying Bellman principle means eliminating from *State*(*i*+1), states (*T*1, *R*1) which are not Pareto optimal, that means which are such that there exists (*T*2, *R*2) in *State*(*i*+1), such that *T*2 ≤ *T*1 and *R*2 ≤ *R*1, one at least of those inequalities being strict.

The Filtering Issue: **A Learning through Reinforcement Device** ([3, 10, 13, 22]). As told in the introduction, SPRC puts computing costs are at stake. Above low level **DP_*Schedule*** procedure should run very fast, and so *State*(*i*), as well as the set *DEC*(*i*) of tried decisions rt^*_e should remain (very) small, and this in spite of the fact that the number of potential vales rt^*_e may be high, and even infinite if we suppose that we are dealing with rational numbers. In order to handle this issue, we use the second part of proposition 4 and the fact that, in perfect cases, rt^*_e might be considered as a **risk per time price**, taking a same *theoretical* value *rt_perfect* all along path Γ. This suggests us to do as if such a theoretical value *rt_perfect* were existing and try to *learn* it through the *Reinforcement Principle* (see [2, 17]), that means while moving along path Γ and performing (possibly several times) our ***BL_RCSP*** algorithm. More precisely:

- We fix the number *M* of possible decisions rt^*_e, and impose a threshold *State_Max* on the size of any state subset *State*(*i*). Those 2 values *M* and *State_Max* become parameters of the ***BL_RCSP*** algorithm.
- According to this, we manage, all along the process, two quantities *rt_min* and *rt_max*, respectively *pessimistic* and *optimistic* estimations of ideal value *rt_perfect*, and which are going to be the target of the learning process. We do in such a way that, for any value *i* during the DP process, decisions rt^*_e which are going to be tried can be written $rt^*_e = rt\_min + m.rt\_max/(M\text{-}1)$, *m* = 0,…, *M*-1.
- Then we drive *rt_min* and *rt_max* values in an auto-adaptative way (*learning through reinforcement*): applying decisions rt^*_e from state subset *State*(*i*) and filtering them through Bellman principle provides us with a state subset *State*(*i*+1) whose size is likely to exceed *State_Max*. Since our interpretation of *rt_min* and *rt_max* values is that *r-midst* = (*rt_min* + *rt_max*)/2 might be considered as the kind of theoretical *risk versus time* price *rt_perfect* we have just been talking above, we rank states (*T*, *R*) of *State*(*i*+1) according to *rt_midst.T* + *R* values. Ideally, states (*T*, *R*) ordered this way should make best states (*T*, *R*) be balanced in the sense that the ratio *R*/*Risk_Max* should be centered around the ratio *TIME*(*o*, x_{i+1})/*TIME*(*o*, *d*) and that the *entropy* of those best states should not be too large. If, for instance, those values are centered significantly above this ratio, then we deduce that we are moving in a too risky way and must make *rt_min* and *rt_max* decrease. Conversely, if those best values are centered above this ratio, then we are too *careful*. More precisely, we perform a kind of statistical analysis of those best values in *State*(*i*+1), and derive, from those *best states* (*T*, *R*), several indicators:
    - *Risk_Balance*: It takes values {*Risky*, *Normal*, *Careful*} depending on the way the mean *R*/*Risk_Max value* is located with respect to *TIME*(*o*, x_{i+1})/*TIME*(*o*, *d*).
    - *Entropy*: It takes values {*Large*, *Normal*, *Small*} depending on the scope of *R*/*Risk_Max* values.

Then ***Clean_Learn*** procedure below performs the *filtering&learning* process:

```
Clean_Learn Procedure:
Rank states (T, R) of State(i+1) according to rt_midst.T + R values;
Select best State_Max (T, R) according to this ranking and compute Risk_Balance and Entropy;
If Risk_Balance = Normal then
        Keep only the State_Max best states in State(i+1);
        If Entropy = Small (Large) then Enlarge (Shorten) the interval [rt_min, rt_max] while keeping rt_midst unchanged;
If Risk_Balance = Risky then
        Split State(i+1) into 2 subsets $S_1$ and $S_2$ with same size: $S_1$ is made of the best states (S, R) according to our ranking and $S_2 = S – S_1$;
        Clean State(i+1) in order to keep the State_Max/2 best states in both $S_1$ and $S_2$;
        Make rt_max and rt_min decrease;
        If Entropy = Small (Large) then Enlarge (Shorten) the interval [rt_min, rt_max] while keeping rt_midst unchanged;
```

If *Risk_Balance* = *Careful* then proceed the same way as in previous case, while making *rt_min* and *rt_max* increase.

C. Greedy Algorithm GR_RCSP

We turn above ***BL_RCSP*** algorithm into a *greedy* one, by removing the top level ***Improve*** loop. That means that we choose Γ as the shortest path from *o* to *d* in the *TIME* sense, and apply ***DP_Schedule***.

IV. A A* LIKE ALGORITHM

Let us first recall that well-know A* algorithm [14] is an extension of Dijsktra algorithm for the search of a shortest path in a graph, which was introduced in order to deal with very large networks. Nodes of such a network are usually defined in an implicit way, as possible configurations for the state of a system (for instance a robot). It is typically our case here, since we are searching a path in a *risk expanded* network, whose nodes are all pairs (*x*, *R*), *x* beings a node of the transit network *G* = (*N*, *A*) and *R* a risk value between 0 and *Risk_Max*.

As in III, we are still willing to design a fast and flexible algorithm. But our approach is different from Section III, in the sense that: Algorithm ***A*_RCSP*** is going to work while simultaneously looking for a path Γ and a schedule *rt** for this path. Roughly, at any time during ***A*_RCSP*** process, we are going to be provided with:

- A current value *T_Curr*, computed by ***GR_RCSP***.
- An *expansion* list *LS* of *state* 3-uples (*x*, *T*, *R*), where:
  - x is a node of the transit network.
  - *T* and *R* are respectively the time and risk value which were required in order to arrive in *x*.
  - *LS* is ordered according to values
    $V = T + TIME(x, d)$. (E2)

  The first element in *LS* is called current *Pivot* state.

  Explanation of (E2): *TIME*(*x*, *d*) is a lower bound for the time that the vehicle *VEH* has to spend running before achieving its journey. So *V* is a lower bound for value *G_Time*(Γ, *rt**) in case routing strategy (Γ, *rt**) extends current position (*x*, *T*, *R*).
- A list *L_PIVOT*, which contains all 3-uple (*x*, *T*, *R*) which have formerly been used a *Pivot* state.

We do in such a way that:

- All elements in *LS* ∪ *L_PIVOT* are Pareto optimal in the sense that, for a given *x*, there does not exist *T*, *R*, *T1*, *R1* such that both (*x*, *T*, *R*) and (*x*, *T*1, *R*1) are in *LS* ∪ *L_PIVOT*, with $T \leq T1$ and $R \leq R1$. (E3)

Then Algorithm ***A*_RCSP*** removes *Pivot* = (x_0, T_0, R_0) from *LS*, puts it into *L_Pivot*, and perform the *expansion* step, that is:

- For any arc *e* = (x_0, *x*), with origin in x_0, it generates a set DEC^e_{Pivot} of decisions rt_e* (*risk speeds*) in the same sense as in ***BL_RCSP*** algorithm; (*DECIDE*)
- For any arc *e* = (x_0, *x*) and any decision rt_e* (*risk speed*) it generates resulting state (*x*, *T*, *R*); Then ***A*_RCSP*** inserts state (*x*, *T*, *R*) into *LS* and manages in such a way that (E2) be satisfied;
- For any *x* such that (x_0, *x*) is an arc of the transit network, it filters states (*x*, *T*, *R*) which are currently in *LS*, in order to meet requirement (E3). (*FILTER*)

The Filtering Issue: ***Learning through Reinforcement.*** Once again, since we want our algorithm to run fast, we impose a threshold *M* on the number of possible decisions rt_e*. Besides, we impose a parameter *State_Max*, with the meaning: (E4)

- For any *x*, the number of states (*x*, *T*, *R*) which are contained into *LS* ∪ *L_PIVOT* never exceeds parameter *Max_State*.

In order to go further with this filtering issue, we must now explain the way instructions *FILTER* and *DECIDE* work, since we see that, as for the ***BL_RCSP*** algorithm, the key point in this ***A*_RCSP*** algorithm lies on the way we perform those instructions. Since we are provided with a current solution *T_Curr*, we may of course apply standard Branch/Bound filtering technique, and kill candidate state (*x*, *T*, *R*) if related value $V = T + TIME(x, d) \geq T\_Curr$. But it is clearly not enough in order to ensure that (E4) is satisfied. So we proceed the same way as in the case of ***BL_RCSP***:

- At any time during the process, we are provided with two quantities *rt_min* and *rt_max*;
- Then we use *rt_min* and *rt_max* in order to generate (Instruction *DECIDE*) decisions *rt** exactly as in ***BL_RCSP***:
  - We rank candidate states (*x*, *T*, *R*), resulting from all decisions $rt*_e$, *e* = (x_0, *x*) as in III.2, while using *rt_midst* = (*rt_max* + *rt_min*)/2.
  - We compute the *Risk_Balance* and *Entropy* quantities.
  - Then we update values *rt_min* and *rt_max* as in III.2, and apply the same technique in order to ensure that, for any *x*, the number of states (*x*, *T*, *R*) in *LS* ∪ *L_PIVOT* does not exceed *State_Max*.

V. NUMERICAL EXPERIMENTS

Goal: We have been performing numerical experiments with the purpose of getting information about the following points:

- The ability of the different algorithms to get good solutions under small computational costs, and the dependence of their behavior to the size of the transit network;

- The sensitivity of those algorithms to the parameter *State_Max* and *M*, which bounds, for every algorithm, the numbers of possible states and decisions;
- The sensitivity of our algorithms to the structure of the piecewise constant functions Π^e, and on the intensity of current traffic inside the transit network at the time when the algorithms are applied.

In order to do it, we used the *A** like algorithm, run with large *State_Max* and *M* values as an almost exact algorithm, which provided us with reference results.

Technical Context: Algorithms were implemented in C++, on a computer running Windows 10 Operating system with an IntelCore i5-6500@3.20 GHz CPU, 16 Go RAM and Visual Studio 2017 compiler.

Instances: We generated networks (*N*, *A*) as connected symmetric *partial grids*, which means grids *n*m*, modified through removal of a percentage ρ of nodes and arcs and the introduction of one-way arcs (we break the symmetry of the grid). Those partial grids are summarized through their number $|N|$ of nodes and their number $|A|$ of arcs. The time value $TIME_{x,y}$ of any arc (*x*, *y*) is 1, as well as related *v_max* value. Function Φ is taken as function u -> $\Phi(u) = u^2$. Function Π^e are generated by fixing a time horizon *T_Max*, fixing a mean number *B* of *break points* t_i^e in [0, *T_Max*] per time unit, and an average value Δ for value Φ(*u*). Then, for any *e*, we randomly generate *break points* t_i^e and values $\Pi^e(t_i^e)$, while imposing those values to belong a finite 5 values set {2Δ, 3Δ/2, Δ, Δ/2, 0}. Finally, we fix the threshold *Risk_Max* value. $TIME_{o,d}$ is also a parameter.

We present here results for 10 instances, whose characteristics come as follows:

TABLE I.
CHARACTERISTICS OF THE INSTANCES

Instance	$\lvert N\rvert$	$\lvert A\rvert$	***B***	***Δ***	***Risk_Max***	$TIME_{o,d}$
1	22	65	1	0.2	1	6
2	18	61	2	0.6	1	7
3	19	65	3	1	1	5
4	54	159	1	0.2	2	9
5	58	182	2	0.6	2	9
6	51	175	3	1	2	8
7	88	285	1	0.2	3	12
8	92	268	2	0.6	3	11
9	83	250	2	0.6	3	10
10	86	262	3	1	3	11

Outputs: For every instance we compute:

- The Risk value *R_BL*, the Time value *T_BL* computed by the bi-level ***BL_RCSP*** algorithm, the number of iterations *ITER* of its main loop (modification of *Γ*) and related CPU time *Time_BL*.
- The Risk value *R_A**, the Time value *T_A** computed by the ***A*_RCSP*** algorithm, the number *Node* of visited nodes and related CPU time *Time_BL*.
- The Risk value *R_GR*, the Time value *T_GR* computed by the greedy algorithm ***GR_RCSP*,** together with related CPU time *CPU_GR*.
- Almost exact Risk value and Time value *R_Opt*, *T_Opt* computed by the ***A*_RCSP*** algorithm, performed with large *State_Max* and *M* values, together with related CPU time *CPU_Opt*.

Obtained results are summarized in the following tables: CPU times are in seconds

TABLE 2.
REFERENCE VALUES AND *GR_RCSP* BEHAVIOR (*STATE_MAX* = 10 AND *M* = 5)

Instance	***R_Opt***	***T_Opt***	***CPU_Opt***	***R_GR***	***T-GR***	***CPU_GR***
1	0.98	7.5	459.6	0.91	11.5	0.01
2	0.99	13.0	524.9	0.94	13.8	0.01
3	0.98	15.5	312.4	0.95	18.2	0.01
4	1.97	9.6	988.7	1.78	12.7	0.02
5	1.98	16.9	1044.2	1.92	24.9	0.02
6	1.98	18.4	857.5	1.88	24.1	0.02
7	2.98	11.0	2209.3	2.84	12.4	0.03
8	2.96	16.7	1858.0	2.81	22.7	0.03
9	2.99	18.9	1977.8	2.80	28.6	0.03
10	2.98	37.5	2033.5	2.68	54.2	0.03

TABLE 3.1.
BL_RCSP BEHAVIOR WITH *STATE_MAX* = 10 AND *M* = 5

Instance	***R_BL***	***T-BL***	***CPU_BL***	***ITER***
1	0.92	8.3	0.05	3
2	0.93	14.0	0.02	1
3	0.91	17.1	0.03	2
4	1.84	10.8	0.05	2
5	1.90	17.7	0.14	5
6	1.82	20.7	0.11	4
7	2.74	12.2	0.08	2
8	2.80	19.4	0.13	3
9	2.87	19.5	0.35	8
10	2.78	40.5	0.24	6

TABLE 3.2.
BL_RCSP BEHAVIOR WITH *STATE_MAX* = 50 AND *M* = 10

Instance	***R_BL***	***T-BL***	***CPU_BL***	***ITER***
1	0.96	8.0	0.64	3
2	0.95	13.7	0.79	3
3	0.95	16.3	0.93	4
4	1.88	10.4	0.59	2
5	1.91	17.5	1.3	4
6	1.92	19.4	0.11	5
7	2.87	11.9	2.1	4
8	2.90	18.6	1.5	4
9	2.89	19.4	2.9	6
10	2.85	39.5	3.6	7

TABLE 4.1
*A*_RCSP* BEHAVIOR WITH *STATE_MAX* = 10 AND *M* = 5

Instance	***R_A****	***T_A****	***CPU_A****	***Node***
1	0.95	7.9	0.15	11
2	0.92	13.8	0.13	10
3	0.97	15.9	0.14	10
4	1.80	10.9	0.35	20
5	1.88	17.7	0.39	18
6	1.86	20.0	0.45	19
7	2.75	11.9	0.95	25
8	2.66	18.0	1.0	28
9	2.80	19.5	1.1	30
10	2.84	38.6	1.2	28

TABLE 4.2
*A*_RCSP* BEHAVIOR WITH *STATE_MAX* = 50 AND *M* = 10

Instance	***R_A****	***T_A****	***CPU_A****	***Node***
1	0.98	7.7	1.2	9
2	0.96	13.6	1.3	10
3	0.98	15.7	1.5	10
4	1.87	10.5	3.2	19
5	1.90	17.5	4.2	17
6	1.93	19.2	4.0	17
7	2.85	11.6	9.8	25
8	2.77	17.6	10.5	28
9	2.88	19.2	10.3	29
10	2.87	38.4	9.9	26

Comments: Results obtained through ***GR_RCSP*** are rather erratic, because this algorithm relies on the current state of shortest path Γ from *o* to *d*, which can be bad at the time when we launch the algorithm. ***A*_RCSP*** tends performs better than ***BL_RCSP*** as for the accuracy, but is more time consuming. Depending on the cases, results may be significantly impacted by parameters values *State_Max* and *M*. Finally, we also notice that obtaining almost exact optimal values is rather time costly, even on small instances. In order to improve it, we should find a way to provide a criterion which could identify, at any times, whether a decision rt^*_e has to be tried or not. Notice also that *Risk_Opt* is almost never equal to *Risk_Max*, in spite of proposition 2, because of the bias due to the discretization of the rt^*_e.

VI. CONCLUSION

We have been dealing here with a shortest path problem with risk constraints, which we handled under the prospect of fast, reactive and interactive computational requirements. But, the true practical problem is supposed to be a *pick up and delivery* one, simultaneously involving several tasks and vehicles. It comes that a future challenge is to adapt the algorithms which we just described here to such a more general PDP context. Also, there exist a demand from industrial players to use those algorithms as a tool for strategic decision, in order to estimate convenient size of the AGV fleet, together with the number of autonomous vehicles inside this fleet. We plan addressing those issues in the next months.

ACKNOWLEDGMENT

Present work was funded by French ANR: National Agency for Research, and Labex IMOBS3, as well as by Region AURA: *Auvergne Rhône Alpes*.

REFERENCES

[1] Amazon.com, inc. *amazon prime ai*r. [online]., Available :http://www.amazon.com/primeair (2013).

[2] C.Artigues, E.Hébrard, A.Quilliot, H.Toussaint: "Models and algorithms for natural disaster evacuation problems". *Proceedings of the 2019 FEDCSIS WCO Conference*, p 143-146, (2019). DOI: http://dx.doi.org/10.15439/978-83-952357-8-8

[3] B. Bakker, S. Whiteson, L. J. Kester, F. Groen: "Traffic light control by multi-agent reinforcement learning systems"; In *Interactive Collaborative Information Systems*, (2010). DOI: 10.1007/978-3-642-11688-9_18

[4] B. Berbeglia, J-F. Cordeau, J-F., I. Gribkovskaïa, G. Laporte: "Static pick up and delivery problems : a classification scheme and survey". *TOP: An Official Journal of the Spanish Society of Statistics and Operations Research* 15, p 1-31, (2007). DOI: 10.1007/s11750-007-0009-0

[5] L. Chen, C. Englund: "Cooperative intersection management: a survey"; *IEEE Transactions on Intelligent Transportation Systems* 17-2, p 570-586, (2016). DOI: 10.1109/TITS.2015.2471812

[6] D. Duque, L.Lozano, A.L.Medaglia. An exact method for the biobjective shortest path problemfor large-scale road network. EJOR 242, p 788-795, (2015). http://dx.doi.org/10.1016/j.ejor.2014.11.003

[7] S.Fidanova, O.Roeva, M.Ganzha: " Ant colony optimization algorithm for fuzzy transport modelling ". *Proceedings of the 2020 FEDCSIS WCO Conference*, p 237-240, (2020). DOI: http://dx.doi.org/10.15439/978-83-955416-7-4

[8] A. Franceschetti, E. Demir, D. Honhon, T. Van Woensel, G. Laporte, and M. Stobbe. "A metaheuristic for the time dependent pollution-routing problem"; *European Journal of Operational Research*, 259 (3): 972 – 991, (2017). DOI: 10.1016/j.ejor.2016.11.026

[9] S. Bsaybes, A.Quilliot, A.Wagler: "Fleet management for autonomous vehicles using multicommodity coupled flows in time-expanded networks"; *17th International Symposium on Experimental Algorithms (SEA 2018) (LIPIcs)* 103, (2018). DOI: 10.4230/LIPIcs.SEA.2018.25

[10] M.Krzyszton: "Adapative supervison: method of reinforcement learning fault elimination by application of supervised learning". *Proceedings of the 2018 FEDCSIS AI Conference*, p 139-149, (2018). DOI: http://dx.doi.org/10.15439/978-83-949419-5-6

[11] J. Kumar, V. V. Ranga: "Multi-robot coordination analysis, taxonomy, challenge and future scope"; *Journal of Intelligent and Robotic Systems*, 102:10, (2021). https://doi.org/10.1007/s10846-021-01378-2

[12] T. Le-Anh, M. B. De Koster:: "A review of design and control of automated guided vehicle systems" *European Journal of Operational Research*, 171, 1-23, (2006). https://doi.org/10.1016/j.ejor.2005.01.036

[13] Y.Li, E.Fadda, D.Manerba, R.Tadei, O.Terzo: " Reinforcement learning algorithms for online single machine scheduling". *Proceedings of the 2020 FEDCSIS WCO Conference*, p 277-283, (2020). DOI: http://dx.doi.org/10.15439/978-83-949419-5-6

[14] Nilsson, J.: *Artificial Intelligence*; SpringerY, (1982). ISBN 978-3-540-11340-9

[15] Philippe, C., Adouane, L., Tsourdos, A., Shin, H.S., Thuilot, B. : "Probability collective algorithm applied to decentralized coordination of autonomous vehicles"; *2019 IEEE Intelligent Vehicles Symp*., 1928–34. IEEE, Paris (2019). DOI:10.1109/IVS.2019.8813827

[16] V. Pimenta, A. Quilliot, H. Toussaint, D. Vigo: "Models and algorithms for reliability oriented DARP with autonomous vehicles";

European Journ. of Operat. Res., 257, 2, p 601-613, (2016). doi: 10.1016/j.ejor.2016.07.037.

[17] Y. Rizk, M. Awad, E. Tunstel: "Cooperative heterogenous mutlti-robot systems: a survey"; *ACM Computing Surveys* 29, (2019). https://doi.org/10.1145/3303848

[18] C. Ryan, F. Murphy, F., Mullins, M.: "Spatial risk modelling of behavioural hotspots: Risk aware paths planning for autonomous vehicles"; *Transportation Research A* 134, p 152-163 (2020).

[19] DOI: 10.1016/j.tra.2020.01.024

[20] K.Stoilova, T.Stoilov: "Bi-level optimization application for urban traffic management". *Proceedings of the 2020 FEDCSIS WCO Conference*, p 327-336, (2020). DOI: http://dx.doi.org/10.15439/978-83-949419-5-6

[21] K. C. Vivaldini, G. Tamashiro, J. Martins Junior, M. Becker: "Communication infrastructure in the centralized management system for intelligent warehouses". *In: Neto, P., Moreira, A.P., et al. (eds.) WRSM 2013*. CCIS, vol. 371, pp. 127–136. Springer, (2013)

[22] I. F. Vis: "Survey of research in the design and control of AGV systems". *European Journal of Operations Research 170:677–709*, (2016). DOI: 10.1016/j.ejor.2004.09.020

[23] J. Wojtuziak, T. Warden, O. Herzog: "Machine learning in agent based stochastic simulation: Inferential theory and evaluation in transportation logistics"; *Computer and Mathematics with Applications* 64, p 3658-3665, (2012).

[24] https://doi.org/10.1016/j.camwa.2012.01.079

[25] M. Zhang, R. Batta, R. Nagi R (2008): "Modeling of workflow congestion and optimization of flow routing in a manufacturing/warehouse facility". *Management Sciences* 55:267–280, (2008). DOI: 10.1287/mnsc.1080.0916

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 163–172
DOI: 10.15439/2021F112
ISSN 2300-5963 ACSIS, Vol. 25

Achieving Good Nash Equilibrium by Temporal Addition of Dummy Players

Ofek Dadush and Tami Tamir
School of Computer Science
The Interdisciplinary Center
Herzliya, Israel
Emails: ofek.dadush@post.idc.ac.il, tami@idc.ac.il

***Abstract*—We consider cost-sharing games in which resources' costs are fairly shared by their users. The total players' cost in a Nash Equilibrium profile may be significantly higher than the social optimum. We compare and analyze several methods to lead the players to a good Nash Equilibrium by temporal addition of dummy players. The dummy players create artificial load on some resources, that encourage other players to change their strategies.**

We show that it is NP-hard to calculate an optimal strategy for the dummy players. We then focus on symmetric singleton games for which we suggest several heuristics for the problem. We analyze their performance distinguishing between several classes of instances and several performance measures.

I. Introduction

In resource allocation applications, a centralized authority is assigning the clients to different resources. For example, in job-scheduling applications, jobs are assigned to servers to be processed; in communication or transportation networks, traffic is assigned to network links to be routed. The centralized utility is aware of all clients' requests and determine the assignment. Classical computational optimization problems study how to utilize the system in the best possible way. In practice, many resource-allocation services lack a central authority, and are often managed by multiple strategic users, whose individual payoff is affected by the assignment of other users. As a result, game theory has become an essential tool in the analysis of resource-allocation services. In the corresponding game, every client corresponds to a selfish player who aims at maximizing its own utilization. Naturally, suboptimal players will keep changing their strategy, and the dynamic continues as long as the profile is not stable. Pure Nash equilibrium (NE) is the most popular solution concept in games. A strategy profile is a NE if no player has a beneficial deviation.

It is well known that decentralized decision-making may lead to sub-optimal solutions from the point of view of the society as a whole. On the other hand, the system cannot control the decisions made by the players. In this work we propose to analyze *the power of adding dummy players* controlled by the system. The goal of the dummy players is to direct the players to a high quality solution, while still keeping their freedom to act selfishly and select their own strategy.

The addition of dummy players is temporal, that is, the final configuration consists of the initial set of players. Since the final configuration must be stable, the goal is to lead the players to a good Nash Equilibrium.

Many real life applications can benefit from adapting this approach. For example, navigation apps users receive information about the current status of the traffic and act accordingly, the provider can adjust the information presented to the users in favor of improving the balancing done on cars and roads (and by that avoid creation of traffic jams). Similarly, in communication networks, the delay of using a link can be artificially increased in order to encourage users to use alternative links.

A. Notation and Problem Statement

For an integer $n \in \mathbb{N}$, let $[n] = \{1, \ldots, n\}$. A *network-formation game* (NFG, for short) [1] is $\mathcal{N} = \langle N, G, \langle s_j, t_j \rangle_{j \in [n]} \rangle$, where N is a set of n players, $G = \langle V, E, c \rangle$ is a weighted graph, and for each $j \in [n]$, the pair $\langle s_j, t_j \rangle$ describes the objective of Player j, namely forming a path from its source vertex $s_j \in V$ to its target vertex $t_j \in V$.

A pure *strategy* of a player $j \in N$ is a path from s_i to t_i. A *profile* in $\mathcal{N}$ is a tuple $p = \langle p_1, \ldots, p_n \rangle$ of strategies for the players, that is, p_j is a path from s_j to t_j. Consider a profile p. Recall that c maps each edge to a cost, intuitively standing for the cost of its formation. The cost of an edge is shared equally by the players that uses it. The players aim at fulfilling their objective with minimal cost.

For a profile p, let $n_e(p)$ denote the load on edge e in p, that is, the number of players that include e in their path. The cost of player j in profile p is defined to be

$$cost_j(p) = \sum_{e \in p_j} c_e / n_e(p).$$

The cost of a profile p is the total players' cost, that is $cost(p) = \sum_{j \in N} cost_j(p)$.

For a profile p and a strategy p_j of player $j \in [n]$, let $[p_{-j}, p'_j]$ denote the profile obtained from p by replacing the strategy for Player j by p'_j. Given a strategy profile p, the *best response* (BR) of player j is $BR_j(p) = \arg\min_{p'_j \in P_j} cost_j(p'_j, p_{-j})$; i.e., the set of strategies that minimize player j's cost, fixing the strategies of all other players. Player j is said to be *suboptimal* in p if it can reduce its cost by a unilateral deviation, i.e., if $p_j \notin BR_j(p)$. If no player is suboptimal in p, then p is a *Nash equilibrium* (NE).

IEEE Catalog Number: CFP2185N-ART ©2021, PTI

Given an initial strategy profile p^0, a BR-sequence from p^0 is a sequence $\langle p^0, p^1, \ldots\rangle$ in which for every $T = 0, 1, \ldots$ there exists a player $j \in N$ such that $p^{T+1} = (p'_j, p^T_{-j})$, where $p'_j \in BR_j(p^T_{-j})$. We restrict attention to games in which such best-response dynamics (BRD) are guaranteed to converge to a NE.

A game $\mathcal{N}$ with k dummy players is an extension of $\mathcal{N}$ into $\mathcal{N}' = \langle N, G, \langle s_j, t_j\rangle_{j\in[n]}, k\rangle$. The dummies have no reachability objective of their own, and are controlled by the system. Every dummy player is assigned on a single edge and increases the load on it, thus, making it more attractive for the other players. Practically, a profile of the game $\mathcal{N}'$ is given by the strategies of N and the location of the k dummies. The dummy players are added to a given initial profile p^0. Due to their addition, some of the players will become suboptimal, and a BR-sequence will be initiated. The system can control which suboptimal player is selected to perform its BR. After a finite number of BR-steps, the dummy players leave the network, and the players may continue the BR-sequence until convergence to a NE.

It is well known that NE profiles may be sub-optimal. Let $OPT(G)$ denote the social optimum of a game $\mathcal{N}$, that is, the minimal possible social cost of a feasible assignment of N, i.e., $OPT(\mathcal{N}) = \min_p cost(p)$. The inefficiency incurred due to self-interested behavior is quantified according to the *price of anarchy* (PoA) [11], [15] and *price of stability* (PoS) [1] measures. The PoA is the worst-case inefficiency of a pure Nash equilibrium, while the PoS measures the best-case inefficiency of a pure Nash equilibrium. Formally, PoA($\mathcal{N}$) $= \max_{p\in NE(\mathcal{N})} cost(p)/OPT(\mathcal{N})$, and PoS($\mathcal{N}$) $= \min_{S\in NE(\mathcal{N})} cost(p)/OPT(\mathcal{N})$.

The goal of the dummy addition is to initiate a BR-sequence in which the players converge to a NE whose cost is as close as possible to the cost of the best NE.

Some of our results refer to symmetric singleton games. These games fit several practical environments such as scheduling on parallel machines, or routing on parallel links [11]. A network formation game that corresponds to a symmetric singleton game is given by m parallel $(s-t)$-links $(e_1, \ldots, e_m)$ and a vector of positive link costs $(c_1, \ldots, c_m)$, where c_i is the activation cost of link i. All the players have the same objective – a path from s to t, and thus, the symmetric strategy space is simply the set of edges. A profile p of the game is given by a vector of loads $(n^p_1, \ldots, n^p_m)$, where n^p_i is the number of players on e_i in profile p. Let $n = \sum_i n^p_i$. We assume, w.l.o.g., that $c_1 \leq c_2 \leq \ldots \leq c_m$. Clearly, the social optimum profile of such a game is simply assigning all the players on the cheapest link e_1. On the other hand, it is well known that the price of anarchy is n even for a simple network with only two parallel links having costs $c_1 = 1$ and $c_2 = n$. Indeed, if all the players are assigned on e_2, then each of them pays $n/n = 1$ and would not benefit from deviating to e_1. Note that for this network, a single dummy assigned on e_1 is sufficient to encourage the players to deviate to e_1.

B. Related Work

Many modern systems provide service to multiple strategic users, whose individual payoff is affected by the decisions made by other users of the system. As a result, non-cooperative game theory has become an essential tool in the analysis of this kind of systems, in particular, routing in networks and job scheduling systems [11], [19], [3], [8], [2], [1].

The addition of dummy players will make some of the players suboptimal, and will cause them to change their strategy. Other player will act in response. Thus, our work is closely related to the study of best-response dynamics. The analysis of BR dynamics consists of three main directions: The first studies whether BR dynamics converge to a NE, if one exists (e.g., [13], [8] and references therein). It is well known that BR dynamics does not always converge to a NE, even if one exists. However, for the class of finite *potential games* [16], [14], a pure NE always exists, and BR dynamics is guaranteed to converge to one. The second direction explores how fast it takes until BR dynamics converges to a NE, e.g., [1], [4], [9]. For some games, such as network formation games, the convergence time may be exponential, while for some games, such as singleton congestion games, fast convergence is guaranteed. The third direction studies how the quality of the resulting NE is affected by the choice of the deviating player. Specifically, the order in which players are chosen to perform their best response moves is crucial to the quality of the equilibrium reached [5].

Other related work deal with games in which some of the players are not selfish. In Stackelberg games [17], [10], [6], [7], a centralized authority selects a fraction of players, denoted *leaders*, and assigns them to appropriately selected strategies, this is called the *Stackelberg strategy*. Each of the remaining players, denoted *followers*, selects its strategy selfishly trying to minimize its cost. The behavior of selfish players leads to a *Stackelberg Nash equilibrium* in which none of the selfish players has a beneficial migration.

The goal is to design Stackelberg strategies that will lead the players to a high quality NE. In [17], it is shown that finding an optimal Stackelberg strategy in job scheduling games is NP-hard, and approximation algorithms are presented. In congestion games on parallel links network the usage of a centrally controlled player can lead to the network optimum if its weight is above certain threshold [10]. In parallel networks, under some constraints, there are even optimal Stackelberg strategies [12].

Our model differs from Stackelberg games as we do not assume that some players obey the system. That is, all the players act selfishly. The added dummy players are temporal, and the system should reach a NE after they vanish. The idea of adding a temporal dummy player in order to change the final equilibrium was first presented in [18]. The paper analyzes the potential *damage* a single dummy player can cause to the social optimum in job scheduling games with weighted players.

C. Our Results

Let p^* be the cheapest NE profile. By assigning a sufficiently large number of dummies on the paths in p^*, these paths would become attractive enough, so that the BR of every player would be to join its path in p^*. Since p^* is a NE, the players will remain on these paths after the dummy players depart. Thus, if the number of dummy players is not limited, then it is possible to guarantee convergence to the best NE. We consider two problems:

1) What is the minimal number of dummy players required to reach the best NE?
2) Given a budget of k dummy players, what is the minimal cost NE that can be reached?

A solution for each of these problems involves also an algorithm for utilizing the dummy players. Specifically, for every profile on the BR-sequence, the algorithm should decide (i) on which links the dummy players are assigned, and (ii) which suboptimal player is activated next to perform its best-response.

In section II we prove NP-hardness of both problems for general networks. Specifically, we present a game with two NE profiles, p^* and p such that $cost(p)/cost(p^*) = \Theta(n)$, it is NP-hard to utilize two dummy players in a way that leads the players to p^*, while it is straightforward to do it with three dummy players.

In Section III we define formally the game on m parallel links and provide several basic observations and properties of BR-sequences. In Section IV we present our heuristics for convergence into the social optimum. In section V we presents our heuristics for a given number of dummies, and in Section VI we presents our experimental results.

The addition of dummy players is one temporal perturbation of a game. We conclude in Section VII where we introduce additional perturbation and suggest some directions for future work. Due to space constraints, some of the proofs and experimental results are omitted from this manuscript.

II. Hardness Proof for General Networks

Let $p^\star$ be a min-cost NE profile. By assigning a large enough number of dummy players on the edges of $p^\star$, it is clearly possible to attract the players to $p^\star$. We show that calculating the minimal number of dummies required for this task is NP-hard. Our hardness proof is based on the hardness proof in [5] that considers a problem of determining the order according to which players perform BRD (Best response dynamics).

Theorem 2.1: The problem of leading the players to the lowest cost NE using the minimal number of dummies is NP-hard.

Proof: Given a game, an initial NE strategy profile, and a value k, the associated decision problem is whether k dummies are sufficient to lead the players to the lowest cost NE. We show a reduction from the *Partition* problem: Given a set of numbers $\{a_1, a_2, ..., a_n\}$ such that $\sum_{i \in [n]} a_i = 2$, where $\forall_{i \in [n]} a_i < 1$, the goal is to find a subset $I \subseteq [n]$ such that $\sum_{i \in I} a_i = \sum_{i \in [n] \setminus I} a_i = 1$. Given an instance of *Partition*, consider the network depicted in Figure 1, with the following initial strategy profile, p^0 of $4n+2$ players:

- $3n$ *partition players*, i_1, i_2, i_3 for all $i \in [n]$. The objective of every triplet i_ℓ, is a $\langle v_{i-1}, v_i \rangle$-path. For all $i \in [n]$, the three corresponding partition players has two strategies: an upper edge of cost $420a_i$ and a lower edge of cost $300a_i$. In p^0, all the partition players use the upper edges.
- Players $1', 2', \ldots, n'$: n players whose objective is an $\langle s', t' \rangle$-path. These players have two strategies: the edge (s', t') of cost $300n$, and the path through (u_1, u_2), whose cost is $1200 - \epsilon$. In p^0, they all use the edge (s', t').
- Player a whose objective is an $\langle s_a, t_{a,b} \rangle$-path. Player a has two strategies: The upper path, and the path through $v_0, v_1, \ldots, v_n$. In p^0, Player a uses the upper path.
- Player b whose objective is an $\langle s_b, t_{a,b} \rangle$-path. Player b has three strategies: The upper path, the path through $s_a, v_0, v_1, \ldots, v_n$, and the path through the edge (u_1, u_2). In p^0, Player b uses the upper path.

Observe that p^0 is a NE. Specifically, each of the partition player has cost $\frac{420a_i}{3}$ and a deviation to the lower edge will lead to cost $300a_i$, Player a's current cost is $248+\frac{\epsilon}{2}$. Deviating to the path through v_0, would result in cost $\frac{420}{4} \cdot 2 + 204 = 414$. Player b's current cost is $248 + \frac{\epsilon}{2}$. Its alternative would cost $154+414 = 568$ (through v_0) or $1200-\epsilon$ (through u_1). Finally, every player on the lower edge has current cost 300, while its alternative through u_1 costs $1200 - \epsilon$.

The following additional observations limit the possible BRD sequences of the game:

1) Not only that the initial profile is a NE, but it is also stable in the presence of a single dummy.
2) In order to initiate a deviation of a partition player i_ℓ, two dummies should be placed on the lower (v_{i-1}, v_i)-edge, as $100a_i = \frac{300a_i}{3} \leq \frac{420a_i}{3} = 140a_i$.
3) The n players currently on (s', t') would benefit from a deviation only after the edge (u_1, u_2) is utilized by three other player.

The following profile p^* is the minimal cost NE of this game and also its social optimum:

- For every $i \in [n], \ell \in [3]$, the partition player i_ℓ uses the lower (v_{i-1}, v_i) edge. of cost $300a_i$.
- Players $1', 2', \ldots, n'$ are on the path through u_1. use the $1200 - \epsilon$ edge.
- Player a is on the path through v_0 and use the lower edges uses the lower (v_{i-1}, v_i) edges.
- Player b is on the path through u_1.

The social optimum cost is $300 \cdot 2 + 204 + 1200 - \varepsilon = 2004 - \varepsilon$.

The main claim of the reduction is based on the properties presented earlier.

Claim 2.2: Two dummies can guarantee convergence to p^* if and only if a partition of exists.

■

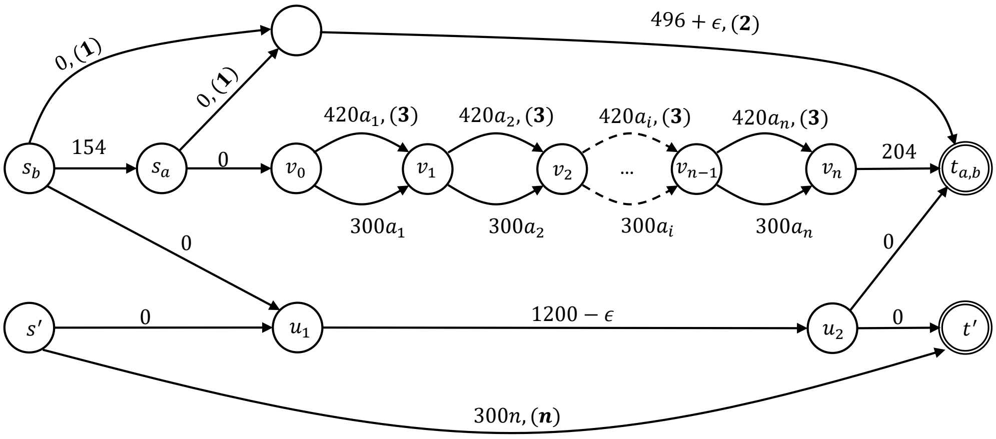


Fig. 1. The network constructed for a given *Partition* instance. Every edge is labeled by its cost, and (in brackets) the number of players using it in p^0.

III. Cost-Sharing Games on Parallel Links

In light of the hardness result for general networks, we consider a network of parallel links. Recall that the network is given by m parallel links $(e_1, \dots, e_m)$ and a vector of positive link costs $(c_1, \dots, c_m)$. A profile p of the game is given by a vector of loads $(n_1^p, \dots, n_m^p)$, where n_i^p is the number of players on e_i in profile p. With fair cost-sharing, the cost of a player assigned on e_i in profile p is c_i/n_i^p. We assume, w.l.o.g., that $c_1 \le c_2 \le \dots \le c_m$. Denote by p_{a^+} the profile obtained from p by adding k dummy players on e_a. Given a profile p, the best response of a player on e_i is denoted $BR_i(p)$. A link $e_j \in BR_i(p)$ if and only if $\frac{c_j}{n_j+1} < \frac{c_i}{n_i}$ and $\forall e_l \ne e_i$, $\frac{c_j}{n_j+1} \le \frac{c_l}{n_l+1}$. In particular, $e_a \in BR_i(p_{a^+})$ if it is possible to attract a player from e_i to migrate to e_a by adding the dummy players on e_a.

Let n_i^0 be the load on e_i in the initial profile p^0. Let e_1 be the cheapest link, were ties are broken in favor of highly loaded links in p^0. That is, for every $i > 1$, either $c_1 < c_i$, or $c_1 = c_i$ and $n_1^0 \ge n_i^0$. Since the game is symmetric, the social optimum cost is c_1.

We present heuristics for solving the following problems: Given a network of parallel links and an initial configuration p^0, (i) what is the minimal number of dummies required to reach the social optimum, and (ii) what is the social lowest cost we can achieve with a given number of dummies. For both problems we assume that the algorithm can move the dummy players among the links, and can select the deviating suboptimal player in each step. Players that get the right to deviate select their best response move.

Performance Measures: Assume that some heuristic is performed on an initial profile p^0. The quality of a solution will be measured by 4 parameters.

1) The Social cost of the final profile.
2) Number of dummies used.
3) Length of BR-sequence till convergence.
4) Number of times the dummy players move.

In our experiments some of these measures are fixed. For example, we tested the social cost achieved by various heuristics with a given number of dummies, or the numbers of dummies required to converge to the social optimum, e_1.

A. Preliminaries and Observations

We start by introducing some notation and stating few important observations and claims.

For a profile p, let $E^p_{min} = \arg\min_{i \in E} \frac{c_i}{n_i^p+1}$, be the set of all most attractive links. Let e^p_{min} be a link in E^p_{min} with a highest cost, breaking ties arbitrarily. Let $price^p_{min} = \frac{c_{e^p_{min}}}{n^p_{e^p_{min}}+1}$ be the cost to be paid by a player that joins the most attractive link.

Observation 3.1: In every NE profile, all the players are assigned on the same link.

Claim 3.2: If $e_a \in BR_{e_b}(p)$ for some link e_b, then we can guarantee convergence of BRD to e_a.

Proof: We show that e_a is the BR as long as the dummy players do not change their location. $e_a \in BR_{e_b}(p)$ if and only if $\frac{c_a}{n_a^p+1} < \frac{c_b}{n_b^p}$ and $\frac{c_a}{n_a^p+1} \le \frac{c_i}{n_i^p+1}$, for every $i \ne b$. After one player moves from e_b to e_a, the load on e_a is $n_a^p + 1$. Now, $e_a \in BR_{e_b}(p^{+1})$ for every $i \ne a$, since

$$\frac{c_a}{n_a^{p^{+1}}+1} = \frac{c_a}{n_a^p+2} < \frac{c_a}{n_a^p+1} \le \frac{c_i}{n_i^p+1} = \frac{c_i}{n_i^{p^{+1}}+1}$$

$$\frac{c_a}{n_a^{p^{+1}}+1} = \frac{c_a}{n_a^p+2} < \frac{c_a}{n_a^p+1} < \frac{c_b}{n_b^p} < \frac{c_b}{n_b^p-1} = \frac{c_b}{n_b^{p^{+1}}}$$

Thus, independent of the order the players are activated, as long as dummy players are not changing their location, every BR sequence converges to e_a. ∎

Observation 3.3: For any $a, b, x, y, k > 0$, if $a \le x$ and $\frac{a}{b} \le \frac{x}{y}$ then $\frac{a}{b+k} \le \frac{x}{y+k}$.

Observation 3.3 implies that if a link is less attractive than e_1, then it will never get players during a sequence that converges to e_1, since it requires at least the same number of dummies as making e_1 the BR of some link directly.

Our next claim states that if a link, e_b, is a best-response of some player, then it is also a best-response of the players on e^p_{min}. Note that if the link e^p_{min} is empty then it must be that $e^p_{min} = e_1$ and convergence to e_1 is possible even without dummy players.

Claim 3.4: For a given profile p and a link $e_b \neq e^p_{min}$, if $\exists e_a$ s.t $e_b \in BR_a(p)$ then $e_b \in BR_{e^p_{min}}(p)$.
In addition, if e_a is the BR of some link in E^p_{min} it is the BR of every link in E^p_{min}.

Claim 3.5: If $e_a \in BR_{e^p_{min}}(p)$ then for every $e_i \in E^p_{min}$, it holds that $e_a \in BR_{e_i}(p)$.

Proof: If $|E^p_{min}| = 1$ then it is clearly true. Else $|E^p_{min}| > 1$ and let $e_u, e_v \in E^p_{min}$, assume $e_a \in BR_{e_u}(p)$. It must be that $\frac{c_a}{n_a+1} < \frac{c_u}{n_u}$ and $\frac{c_a}{n_a+1} \leq \frac{c_v}{n_v+1} < \frac{c_v}{n_v}$. Since $\frac{c_u}{n_u+1} = \frac{c_v}{n_v+1}$, we conclude $e_a \in BR_{e_v}(p)$ for any other link in E^p_{min}. ■

B. The Naive Solution

Before presenting the more complicated heuristics, we present a naive solution that is based on directly making e_1 the best-response of some player. By Claim 3.2, once e_1 is the BR of some link, we can guarantee convergences to e_1, and based on Claim 3.4, we can calculate the number of dummies required to directly make e_1 the BR of e^p_{min}.

Recall that $e_1 \in BR_{e^p_{min}}(p_{1+})$ if and only if $\frac{c_1}{n^p_1+k+1} < \frac{c_{e^p_{min}}}{n^p_{e^p_{min}}}$, and $\frac{c_1}{n^p_1+k+1} \leq \frac{c_i}{n^p_i+1}$, for every $i \neq e^p_{min}$. Therefore, the minimal integer k satisfying

$$\frac{c_1}{n_1+k+1} < \frac{c_{e^p_{min}}}{n^p_{e^p_{min}}} \text{ and } \frac{c_1}{n_1+k+1} \leq \min_{i \neq e^p_{min}} \frac{c_i}{n_i+1}$$

is the minimal number of dummies required to directly make e_1 a BR of some player. We get that k is the minimal integer satisfying

$$k > c_1 \cdot \frac{n^p_{e^p_{min}}}{c_{e^p_{min}}} - (n_1+1) \text{ and } k \geq c_1 \cdot \max_{i \neq e^p_{min}} \left(\frac{n_i+1}{c_i}\right) - (n_1+1) \tag{1}$$

Denote by $k_{naive}(p)$ the minimal k satisfying 1. Specifically, $k_{naive}(p^0)$ is the number of dummies required by the naive solution.

Table I presents an instance demonstrating that the naive solution is suboptimal. Moreover, the number of dummies it needs is higher by factor of about 1.5 from the optimum. The network consists of 8 links whose costs are listed in the first row. The loads in the initial profile p^0 are listed in the second row. The additional rows specify for each link the cost per player in p^0 – given by $\frac{c_e}{n^0_e}$, and cost per player if one player joins e, given by $\frac{c_e}{n^0_e+1}$.

Links $e_2 - e_7$ all have the same cost and initial load.

We can easily see that without dummies, regardless of the activation order, the players will converge into e_8, which is the most expensive link. Using the naive solution, the required number of dummies needed to converge into e_1 is $k_{naive}(p^0) = 300$. We show that convergence to e_1 can be achieved using $k = 220$ dummies.

Link	e_1	$e_2 - e_7$	e_8
Cost	3000	3100	6000
Load	0	200	600
$\frac{c_e}{n^0_e}$	- -	15.5	10
$\frac{c_e}{n^0_e+1}$	3000	15.42	9.98

TABLE I
INITIAL PROFILE p^0 OF INSTANCE I_1

By assigning 220 dummies on e_2 and activating a player on e_8, a migration from e_8 to e_2 will be performed, then we assign the dummies on e_3 and activate a player on e_8 which creates a migration from e_8 to e_3, we then continue in a round robin fashion on links $e_2 - e_7$ until 159 players leave e_8 and the following profile is reached:

Link	e_1	$e_2 - e_4$	$e_5 - e_7$	e_8
Cost	3000	3100	3100	6000
Load	0	227	226	441
$\frac{c_e}{n^p_e}$	- -	13.65	13.71	13.60
$\frac{c_e}{n^p_e+1}$	3000	13.59	13.65	13.57

TABLE II
THE PROFILE ACHIEVED AFTER PHASE 1

The BR-sequence proceed after the 220 dummies are moved to e_1 and a player on e_8 is activated. Since the cost of a player who would join e_1 is $\frac{3000}{0+220+1} = 13.57$, a player on e_8 will choose e_1 as its BR and by Claim 3.2 convergence to e_1 is guaranteed. We conclude that convergence to e_1 can be achieved with only 220 dummy players, while the naive solution requires 300 dummies.

IV. Convergence to the Social Optimum

In this section we present our heuristics for convergence into the best NE. In a network of parallel links, the best NE is also the social optimum and is simply e_1, the cheapest edge in the network.

In Observation 3.3 we showed that making links that are less attractive than e_1 the BR of some link is at least as demanding as making e_1 the BR of the same link, therefore, in all our heuristics we do not use such links as BR of any link. Furthermore, for a given instance I, let I' be an instance with the same set of links and load vector in which $n'_i \geq n_i$ for links fulfilling $e_i \in BR_{e^p_{min}}(p_{i+})$. Intuitively, I' is more challenging than I since the links that are more attractive than e_1 become even more attractive. Such links will also not be used in the heuristics we present.

The heuristics we present consists of two phases. In the first phase, players are encouraged to migrate such that the players' cost on the links that are more attractive than e_1 is more balanced compared to p^0, and then apply the naive solution on the more balanced profile. The goal is to use fewer than $k_{naive}(p^0)$ dummies for the balancing phase, as well as to reach a profile p where $k_{naive}(p) < k_{naive}(p^0)$.

A. Max Cost-reduction Heuristic

The first heuristic we present balances the players costs on links that are more attractive than e_1 by migrating players out of the most attractive link, e^p_{min}, into a link that will gain a maximal cost-reduction by an addition of one player. Formally, a link for which $\frac{c_i}{n_i} - \frac{c_i}{n_i+1}$ is maximal. Intuitively, we want the migration to be as significant as possible.

The algorithm gets as input a profile, p^0, and the number, k, of dummies, and returns a binary indicator stating whether the max cost-reduction heuristic can be used to lead the players to e_1. The minimal number of required dummies, can therefore be computed by binary search in the range $[0, k_{naive}(p^0)]$.

Let e^p_{mcr} be a link for which $\max(\frac{c_i}{n_i} - \frac{c_i}{n_i+1})$ is maximal out of the links that can attract players from e^p_{min}. In every iteration the algorithm moves a player from the current e^p_{min} profile to the current e^p_{mcr} until the profile is balanced enough to enable a migration to e_1 (step 3), or identifying that k dummies are not sufficient as a naive solution from the most balanced prnced profile (this is detected in step 11 - by having a loop in the balanced prncing phase).

Algorithm 1 Max Cost-reduction Heuristic (decision version)

```
 1: repeat
 2:     Calculate $E^p_{min}$. Let $e^p_{min} \in E^p_{min}$ be a link with max
        cost.
 3:     if $e_1 \in E^p_{min}$ or $e_1 \in BR_{e^p_{min}}(p_{1^+})$ then
 4:         return $true$
 5:     else
 6:         Let $e^p_{mcr}$ be a link such that $e_{mcr} \in BR_{e^p_{min}}(p_{e^+_{mcr}})$
            for which $(\frac{c_i}{n_i} - \frac{c_i}{n_i+1})$ is maximal.
 7:         place $k$ dummies on $e^p_{mcr}$.
 8:         activate a player from $e^p_{min}$ (creates a migration from
            $e^p_{min}$ to $e^p_{mcr}$).
 9:         remove $k$ dummies from $e^p_{mcr}$.
10:     end if
11: until loop has been detected (profile $p = p^{-2}$)
12: return $false$
```

When the max cost-reduction heuristic is applied on the instance I_1 presented in Table I and $k = 220$, it is able to reach the social optimum. In fact, the BR-sequence performed is exactly the one described in Table II.

B. balanced prncing Heuristic

The second heuristic we present calculates a target load vector in which the marginal cost on the links are balanced. The dummy players are used to achieve this load vector. The naive algorithm is then performed on the balanced profile. The load vector is a one that maximizes the marginal cost on the most attractive link and the second most attractive link. This way the attractiveness of the competitors of e_1 is as low as possible.

For a profile p, let $c^p_{min1} = \min_{i \in E^p_{min}} \frac{c_i}{n^p_i}$, and let e^p_{min1} be a link determining c^p_{min1}. Also, let $c^p_{min2} = \min_{i \in E \setminus \{e^p_{min1}\}} \frac{c_i}{n^p_i+1}$.

The idea is to balance the load on the links such that the minimal among these two values are maximal. Intuitively, this way, by activating a player on e^p_{min1}, the attractiveness of the competitors of e_1 is as low as possible. Calculating the exact load vector achieving maximal $\min\{c^p_{min1}, c^p_{min2}\}$ is computationally hard. In order to simplify the calculations, we calculate instead p_{bal} - a load vector that approximates the optimal one. p_{bal} is defined in the following way: Let $n_{\bar{1}} = \sum_{i>1} n^0_i$ be the number of players that are not assighed on e_1 in p^0 and let $c_{\bar{1}} = \sum_{i>1} c_i$ be the total cost of edges except e_1. In p_{bal} we determine the assignment of the $n_{\bar{1}}$ players that are not on e_1. We first determine load $\left\lfloor \frac{c_i}{c_{\bar{1}}} \cdot n_{\bar{1}} \right\rfloor$ on every link e_i for $i > 1$, we then add the remaining players iteratively, each time adding a player on a link with maximal $\frac{c_i}{n^p_i+1}$.

For example, the profile p_{pal} of the instance I_1 introduced in Table I is the following.

Link	e_1	$e_2 - e_5$	$e_6 - e_7$	e_8
Cost	3000	3100	3100	6000
Load	0	227	226	440
$\frac{c_e}{n^p_e}$	- -	13.65	13.71	13.63
$\frac{c_e}{n^p_e+1}$	3000	13.59	13.65	13.60

TABLE III
PROFILE p_{bal} OF INSTANCE I_1

Once p_{bal} is calculated, we would like to reach this profile from p^0 using the lowest possible number of dummies. Given p and p_{bal}, let $E^p_{drop} = \{e_i | n^p_i > n^{p_{bal}}_i\}$ be the set of edges whose load is higher than their load in p_{bal} and let $E^p_{gain} = \{e_i | n^p_i < n^{p_{bal}}_i\}$ be the set of edges whose load is lower than their load in p_{bal}. Given p, E^p_{drop} and E^p_{gain}, let $k_{migration}$ be the minimal number of dummies required to achieve a migration from a link $e_a \in E^p_{drop}$ to a link $e_b \in E^p_{gain}$, and let the source and target links be e_{drop} and e_{gain}, respectively. The algorithm iteratively calculates e_{drop} and e_{gain} and perform the corresponding migrations. The number of dummies required may increase during the algorithm, and k is updated accordingly. When k is large enough to enable a migration to e_1, that is, when $k_{migration} \geq k_{naive}(p)$, convergence to e_1 is guaranteed.

C. Exhaustive Heuristic

The third heuristic we consider balances the links that are more attractive than e_1 by migrating players outside of the most attractive link e^p_{min} into the least attractive possible link while making sure $price_{min}$ is not getting lower. Formally for a profile p, let $E^p_t = \{e_a | e_a \in BR_{e^p_{min}}(p_{a^+})$ and $\frac{c_a}{n^p_a+2} > price^p_{min}\}$ be the group of target links, meaning they are BR of e^p_{min} if the dummy player are added on them and do not lower $price_{min}$ if a migration from e^p_{min} to that link occurs. Let e^p_t be the link in E^p_t with maximal $\frac{c_a}{n^p_a+2}$. Our algorithm is based on moving players out of e^p_{min} into e^p_t.

The algorithm gets as input a profile, p^0, and the number, k, of dummies, and returns a binary indicator stating whether the exhaustive heuristic leads the players to e_1. The minimal

Algorithm 2 Balancing Heuristic (min k version)

```
Calculate $p_{bal}$
set $k = 0$
repeat
    Calculate $k_{migration}, e_{gain}, e_{drop}$.
    if $k_{migration} \geq k_{naive}(p)$ then
        return $\max\{k, k_{naive}(p)\}$.
    end if
    $k = \max\{k, k_{migration}\}$ .
    place $k$ dummies on $e_{gain}$
    while $n_i^{p_b} - n_i^p > 0$ do
        activate a player from some $e \in E^p_{drop}$ (creates a migration into $e_{gain}$).
    end while
until $E^p_{gain} = \emptyset$
return $\max\{k, k_{naive}(p)\}$
```

number of required dummies, can therefore be computed by binary search in the range $[0, k_{naive}(p^0)]$.

Algorithm 3 Exhaustive Heuristic (decision version)

```
1:  repeat
2:      Calculate $E^p_{min}$, let $e^p_{min} \in E^p_{min}$ be a link with max cost and let $price^p_{min} = \frac{c_{e^p_{min}}}{n^p_{e^p_{min}}+1}$.
3:      if $e_1 \in E^p_{min}$ or $e_1 \in BR_{e^p_{min}}(p_{1^+})$ then
4:          return $true$
5:      else if $E^p_t \neq \emptyset$ then
6:          place $k$ dummies on $e^p_t$.
7:          activate a player from $e^p_{min}$ (creates a migration from $e^p_{min}$ to $e^p_t$).
8:          remove $k$ dummies from $e^p_t$.
9:      end if
10: until no player has being activated
11: return $false$
```

D. Performance Measure Comparison

Recall that in Section III the performance measures according to which the quality of our heuristics is evaluated is listed.

Table IV describe an initial profile of instance I_2 with 8 links. It is easy to see that BRD would converge into e_6 if no dummies are used. Table V summarizes the strengths and weaknesses of the different heuristics when applied on I_2.

Clearly, the naive solution is most efficient in terms of BR-steps and dummy moves, on the other hand, the number of dummies required to reach e_1 is significantly higher. The max cost-reduction needs more dummies than the other algorithms but dominates the number of steps. The balancing algorithm moves the dummies only once for every link in E_{gain}, so the number of dummy moves is very low. The number of dummies is lower than the naive solution. Finally, the exhaustive heuristic achieves the social optimum using the least number of dummies.

V. Exploit a Given Number of Dummy Players

In this section we present our heuristics for finding the lowest achievable social cost, using a limited budget of k dummies. Notice that if we know that e_1 cannot become the BR of any link using a given K, then, as explained in the previous section, migrating players out of e_1 cannot be helpful.

We modify the algorithms presented in Section IV for the new goal. As we elaborate below, the Naive approach and the balancing heuristic are slightly modified, only their destination link may be more expensive than e_1. The two other heuristics, specifically, max cost-reduction and exhaustive, have a different version for the new goal.

Recall that in the naive solution (see Section III-B) the algorithm locates the dummies on the target link. When the number of dummies is limited, we simply calculate the minimal ℓ such that

$$\frac{c_\ell}{n_\ell + k + 1} < \frac{c_{e^p_{min}}}{n^p_{e^p_{min}}} \text{ and } \frac{c_\ell}{n_\ell + k + 1} \leq \min_{i \neq e^p_{min}} \frac{c_i}{n_i + 1}.$$

The corresponding link e_ℓ, is the solution of the naive algorithm with a budget of k dummies.

Next, we describe how the balancing heuristic is tuned for the budged problem: recall that the idea is to calculate a target load vector and lead the players on $\{e_2, \ldots, e_n\}$ to the corresponding configuration. With a given number of dummies, we calculate the minimal ℓ such that it is possible to balance the players on $\{e_{\ell+1}, \ldots, e_n\}$, thus leading the players to e_ℓ. That is, the original algorithm is applied only on subset of the links. Recall that once e_ℓ is a BR of some link, then all other players would benefit from joining it, in particular, those on $e_1, \ldots, e_{\ell-1}$.

A. Max Cost-reduction Heuristic

Recall that in this heuristic, players are migrated out of e^p_{min} into a link in $BR_{e^p_{min}}$. With a given budget of k dummies we run the same algorithm, and keep track of the lowers-cost link that was a target of some migration during the run.

Therefore, the algorithm differs from the decision version, only in the return statements. In line 4, instead of True it returns e_1 and in line 12, instead of False it returns the minimal i s.t $e_i \in E^p_{min}$ or $e_i \in BR_{e^p_{min}}(p_{i^+})$in any seen p.

B. Exhaustive Heuristic

In its decision version (Algorithm 3), the Exhaustive heuristic is used to decide whether convergence to e_1 is possible with k dummies. We now show that without changing the algorithm we can answer the optimization question, namely, what is the lowest-cost link we can converge to using k dummies.

The algorithm differs from the decision version, only in the return statements. In line 4, instead of True it returns e_1 and in line 11, instead of False it returns the minimal i s.t $e_i \in E^p_{min}$ or $e_i \in BR_{e^p_{min}}(p_{i^+})$.

Based on Claim 3.2, when the algorithm returns i such that $e_i \in E^p_{min}$ or $e_i \in BR_{e^p_{min}}(p_{i^+})$, then we can converge to e_i. We show that links that were the BR of some link, will always be able to attract a player from some link, and that

Link	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
Cost	3000	3100	3100	3100	3100	6000	6000	10000
Load	0	188	180	179	175	700	620	600
$\frac{c_e}{n_e^p}$	- -	16.48	17.22	17.127	17.71	8.57	9.67	16.66
$\frac{c_e}{n_e^p+1}$	3000	16.40	17.127	17.22	17.61	8.55	9.66	16.63

TABLE IV
INITIAL PROFILE OF INSTANCE I_2.

algorithm	Naive	Max CR	Balancing	Exhaustive
# of dummies	350	251	310	240
# of steps	2,642	2,956	2,874	3,043
# of dummy moves	1	312	5	312

TABLE V
RESULTS FOR PROFILE I_2

links that were not the BR of any link until we exhausted our effort to make e_1 the BR of some link, will not be able to be the BR of any link. This implies that the returned link is indeed the best achievable link.

The following claims and observations will be used in the analysis of the Exhaustive Heuristic. The first claim shows that $price_{min}$ is monotonically increasing during the algorithm.

Claim 5.1: $price^{p^{+1}}_{min} \geq price_{min}$

The next claim shows that the load on any link $e_m \in E^p_{min}$ monotonically decreases.

Claim 5.2: For a profile p and $e_m \in E^p_{min}$, n^p_a monotonically decreases after profile p.

We now combine Claims 5.1 and 5.2 to show that, $price_{min}$ strictly increases every $|E^p_{min}|$ iterations and at the worst case after $n-1$ iterations.

Observation 5.3: For a given profile p, $price^{p+|E^p_{min}|}_{min} > price^p_{min}$

The next claim shows that if e_a can attract players in some profile p, then it can attract additional players in any profile p' that succeeds p.

Claim 5.4: For a profile p and profile p^{+i} (where $i > 0$) reached in a later stage of our algorithm. If $e_a \in BR_{e^p_{min}}(p_{a+})$, then either $e_a \in BR_{e^{p^{+i}}_{min}}({p^{+i}}_{a+})$ or $e_a = e^{p^{+i}}_{min}$.

We turn to show that if a link was not the BR of any link at no point of the algorithm it cannot become the BR using k dummies.

Claim 5.5: For a profile p^{+i} $(i \geq 0)$ where not $e_1 \in E^p_{min}$ or $e_1 \in BR_{e^p_{min}}(p_{1+})$ or $E^p_t \neq \emptyset$. If $e_a \notin E^p_{min}$ and $e_a \in BR_{e^p_{min}}(p_{a+})$ then we cannot converge into e_a

Proof: Assume by contradiction that for some profile p, $e_a \in BR_{e^p_{min}}(p_{a+})$ but a profile p^{+i} can be reached where $e_a \notin BR_{e^{p^{+i}}_{min}}({p^{+i}}_{a+})$ and $e_a \neq e^{p^{+i}}_{min}$. Using Claims 5.1 and 5.2, $n^{p^{+i}}_{e^p_{min}} < n^p_{e^p_{min}}$ and $price^{p^{+i}}_{min} \geq price^p_{min}$. Furthermore, if $n^{p^{+i}}_a \geq n^p_a$, meaning he only gained players, then $e_a \in BR_{e^p_{min}}({p^{+i}}_{a+})$ and using Claim 5.3 $e_a \in BR_{e^{p^{+i}}_{min}}({p^{+i}}_{a+})$ or $e_a = e^{p^{+i}}_{min}$.

Else $n^{p^{+i}}_a < n^p_a$, then for some profile p^{+j}, where $0 < j < i$ reached between p and p^{+i} $e^{p^{+j}}_{min} = e_a$. If e_a is still the minimum then clearly $e_a = e^{p^{+i}}_{min}$, else it is not the minimum and as seen in Claim 5.2 $\frac{c_a}{n_a^{p^{+i}+k+1}} < price^{p^{+i}}_{min}$ meaning $e_a \in BR_{e^{p^{+i}}_{min}}({p^{+i}}_{a+})$. ■

VI. EXPERIMENTAL RESULTS

In this section we present some experimental results, achieved by simulating the heuristics presented in Sections IV and V. The heuristics were performed on random instances in random initial profiles.

We created a test-base consisting of four classes. The first class, denoted *random* includes instances with a random number of links (between 6 and 20), for each link there was a randomly generated cost between $2,000 - 100,000$ and load between 0 and $10,000$. All values were drawn assuming uniform distribution in their range. Figure 2 shows that in the majority of the random profiles the heuristics are redundant, but there exist a significant amount of profiles in which they are helpful, in those cases the needed number of dummies decrease significantly.

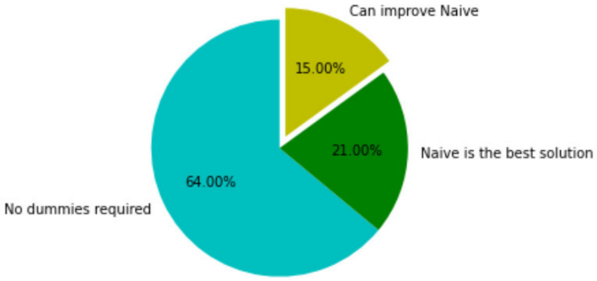


Fig. 2. Percentage of random profiles in which the addition of dummy players is beneficial, and the heuristics are more efficient than the naive solution.

In order to emphasis the differences between the different heuristics, we included in our test-base only instances for which, in at least two heuristics, the required number of dummies is lower than the number of dummies required by the naive solution.

The second class of instances is the most challenging one. It is denoted *"Beat The Competitor"*. In the initial profiles of this class, e_2 has the highest initial load, therefore, the social optimum, e_1, has an attractive competitor.

The third class of instances is denoted *"Beat The Giants"*. In the initial profiles of this class, the initial load on e_1 is very low. There are a few heavily loaded links which cost several times the cost of e_1. Without the use of dummies, any BRD will converge to one of them. In addition there is a larger number of contender links with a lower cost than the heavy links and less attractive than them, but they can attract players from the heavy links with less dummies than e_1.

The last class of instances is denoted *"Beat The Median"*, it is a generalization of the above class. It is similar to *Beat The Giants* with an addition of few links that are even heavier than the heavy links of *Beat The Giants* and the players on those links pay around the same cost as the players on the contender links.

We first present our results for the problem considered in Section IV: what is the number of dummies required to converge to the social optimum. Thus, we fix the social cost of the final profile, and measure the 3 other parameters characterizing the quality of a solution.

Figure 3 shows the number of dummies required to converge to the social optimum scaled compared to the naive solution, and averaged on all instances in the test-base. As shown in the figure, the heuristics can achieve the social optimum with significantly less dummies than the naive solution. In particular, the exhaustive heuristic is always the solution that uses the least number of dummies.

Only In 1% out of the random profiles the balancing algorithm was better than the naive solution. On the other hand, for the classes *Beat The Giants / Median*, the balancing phase is essential.

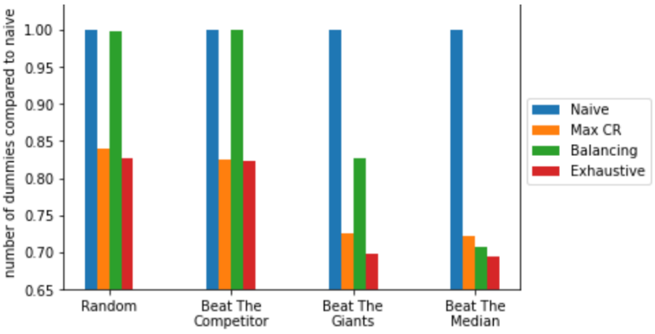


Fig. 3. Number of dummies required in order to converge to the social optimum, compared to the naive solution.

Figure 4 compares the length of the BR-sequence till convergence to e_1, scaled by $n_{\bar{1}}$, which is the number of players that need to migrate to e_1. That is, how many times each player is activated on average. Recall that in the naive solution, every player, except for the players that are assigned on e_1 in p^0, migrates exactly once. Clearly, this is our lower bound. We can see that all the heuristics perform well with respect to this measure. Specifically, even in the longest sequences, players migrate on average at most 1.125 times, but we can clearly see that in the more specific profiles *Beat The Giants / Median* the length of BR-sequence is larger than the naive solution while $Random$ and *Beat The Competitor* lengths on all heuristics are close to the optimal solution.

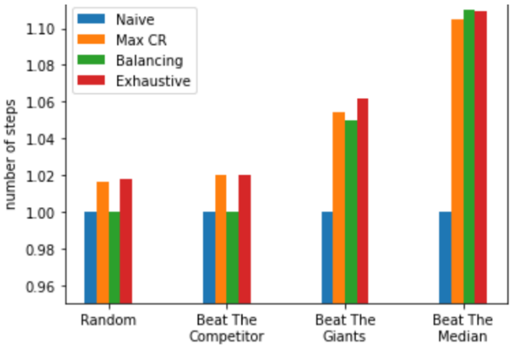


Fig. 4. Length of BR-sequence, measured by the average number of migrations performed by each player that is not on e_1 in p^0.

The next measure we consider is the number of times the dummy players are moved. Again, the naive heuristic provides the lower bound, as the dummy players are assigned exactly once – on e_1. Figure 5 presents the results for this measure, scaled by the number of links in the network. The naive solution assigns the dummy players only once, and the balancing heuristic move the dummies at most $|E_{gain}|$ times. In contrast, the exhaustive and max cost-reduction heuristics may migrate the dummies many times.

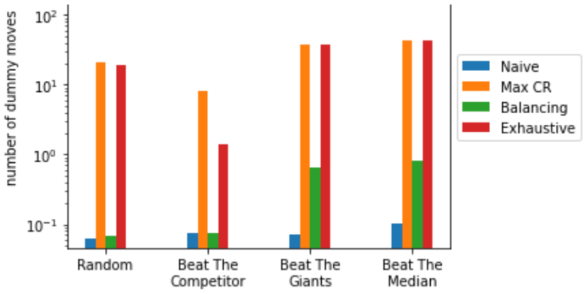


Fig. 5. Number of dummy moves scaled by the number of links

We turn to present our results for the problem considered in Section V: what is the lowest cost achievable link, given a limited budget k of dummy players. Thus, the interesting measure of a heuristic is the resulting social cost.

Figure 6 presents the social costs achieved by each Heuristic where the given k is a parameter which is a percentage of $k_{naive}(p^0)$.(naive is the minimal i we can achieve using one assignment of dummies on some link)

In the chart, the Social cost is divided by the Social Optimum showing how far is the result from the Social optimum.

We can see that in the $Random$ and *"Beat The Competitor"*, the separation from the naive solution is shown in the higher percentage of dummies but in *Beat The Giants / Median* there is a difference even in the lower percentages.

VII. Conclusions and Open Problems

In this work we demonstrated the power of a temporal addition of dummy players to a game. The dummy players

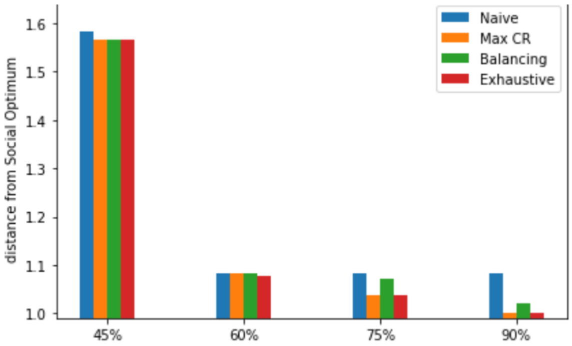


Fig. 6. *"Beat The Giants"* social cost by algorithm and percentage of dummies from the naive solution, divided by social optimum

initiate a dynamic in which the players are encouraged to reach a Nash Equilibrium profile of better quality. We suggested several heuristics for operating the dummies, and analyzed their quality distinguishing between the number of dummy players, the value of the final solution, and the convergence time. Our main message is that the use of dummy players may significantly improve the equilibrium inefficiency. In general, finding an optimal algorithm for exploiting the dummies is NP-hard. However, as we show, even simple heuristics may need significantly less dummies than a naive solution, and the quality of the final solution can improve further by extending the length of the BR-sequence by a factor of 1.15, and if the dummy players can be migrated intensively. Practically, in real-world application such as routing, the possibility of creating a controlled fake load or by temporarily disable the use of some resources, can help migrate the players to routes that improve the global system's performance.

The addition of dummy player is only one possible perturbation of a stable solution. It would be interesting to study the power of additional temporal perturbation in resource allocation games, that refer not only to the set of participating clients, but to set of of resources, e.g., temporal closure or addition of resources or temporal change in resources' activation cost.

References

[1] E. Anshelevich, A. Dasgupta, J. Kleinberg, E. Tardos, T. Wexler, and T. Roughgarden. The price of stability for network design with fair cost allocation. *SIAM Journal on Computing*, 38(4):1602–1623, 2008.

[2] V. Bilò and C. Vinci. On the impact of singleton strategies in congestion games. In *Proc. 25th Annual European Symposium on Algorithms*, pages 17:1–17:14, 2017.

[3] I. Caragiannis, M. Flammini, C. Kaklamanis, P. Kanellopoulos, and L. Moscardelli. Tight bounds for selfish and greedy load balancing. *Algorithmica*, 61(3):606–637, 2011.

[4] E. Even-Dar and Y. Mansour. Fast Convergence of Selfish Rerouting. In *Proc. of SODA*, pp. 772–781, 2005.

[5] M. Feldman, Y. Snappir, and T. Tamir. The efficiency of best-response dynamics. In *The 10th International Symposium on Algorithmic Game Theory (SAGT)*, 2017.

[6] A. Fanelli, M. Flammini, and L. Moscardelli. Stackelberg strategies for network design games. In Proc. of International Workshop on Internet and Network Economics (WINE). pp. 222 –âĂŞ233, 2010.

[7] D. Fotakis. Stackelberg strategies for atomic congestion games. *Theory of Computing Systems*, 47(1):218–249, 2010.

[8] T. Harks and M. Klimm. On the existence of pure nash equilibria in weighted congestion games. *Math. Oper. Res.*, 37(3):419–436, 2012.

[9] S. Ieong, R. McGrew, E. Nudelman, Y. Shoham, and Q. Sun. Fast and compact: A simple class of congestion games. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 2*, AAAI'05, 2005.

[10] Y. A. Korilis, A. A. Lazar, and A. Orda. Achieving network optima using stackelberg routing strategies. *IEEE/ACM Trans. Netw.*, 5(1):161–173, 1997.

[11] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. *Computer Science Review*, 3(2):65–69, 2009.

[12] W. Krichene, J. D. Reilly, S. Amin, and A. M. Bayen. Stackelberg routing on parallel networks with horizontal queues. *IEEE Transactions on Automatic Control*, 59(3):714–727, 2014.

[13] I. Milchtaich. Congestion games with player-specific payoff functions. *Games and Economic Behavior*, 13(1):111 – 124, 1996.

[14] D. Monderer and L. S. Shapley. Potential Games. *Games and Economic Behavior*, 14: 124–143, 1996.

[15] C. H. Papadimitriou. Algorithms, games, and the internet. In *Proc. 33rd ACM Symp. on Theory of Computing*, pages 749–753, 2001.

[16] R. W. Rosenthal. A class of games possessing pure-strategy nash equilibria. *International Journal of Game Theory*, 2:65–67, 1973.

[17] T. Roughgarden. Stackelberg scheduling strategies. *SIAM Journal on Computing*, 33(2):332–350, 2004.

[18] T. Tamir. The power of one evil secret agent. *Theoretical Computer Science*, 839:1–12, 2020.

[19] B. Vöcking. *Algorithmic Game Theory*, chapter 20: Selfish Load Balancing. Cambridge University Press, 2007.

DOI: 10.15439/2021F22
ISSN 2300-5963 ACSIS, Vol. 25

InterCriteria Analyzis of Hybrid Ant Colony Optimization Algorithm for Multiple Knapsack Problem

Stefka Fidanova
Institute of Information and
Communication Technology
Bulgarian Academy of Sciences
Acad. G. Bonchev Str., bl. 25A,
Sofia, Bulgaria
E-mail: stefka@parallel.bas.bg

Maria Ganzha
System Research Institute
Polish Academy of Sciences
Warsaw, Poland
E-mail: maria.ganzha@ibspan.waw.pl

Olympia Roeva
Institute of Biophysics and
Biomedical Engineering
Bulgarian Academy of Sciences
Acad. G. Bonchev Str., bl. 105,
Sofia, Bulgaria
E-mail: olympia@biomed.bas.bg

***Abstract*—The local search procedure is a method for hybridization and improvement of the main algorithm, when complex problems are solved. It helps to avoid local optimums and to find faster the global one. In this paper we apply InterCriteria analysis (ICrA) on hybrid Ant Colony Optimization (ACO) algorithm for Multiple Knapsack Problem (MKP). The aim is to study the hybrid algorithm behavior comparing with traditional ACO algorithm. Based on the obtained numerical results and on the ICrA approach the efficiency and effectiveness of the proposed hybrid ACO, combined with appropriate local search procedure are confirmed.**

***Index Terms*—Local Search, Ant Colony Optimization, InterCriteria analysis, Knapsack Problem**

I. Introduction

ENGINEERING applications normally lead to complex decision make problems. Large scale problems can not be solved with traditional numerical methods. It is a challenge to develop a new techniques, which have simple structure and easy application, and can find near optimal solution even the information about the problem is incomplete. In most of the cases these problems are HP-hard.

Nature inspired methods are more appropriate for solving NP-hard optimization problem, than other methods, because they are flexible and use less computational resources. They are base on stochastic search. The most popular methods are Evolutionary algorithm [28], [45], which simulates the Darwinian evolutionary concept, Simulated Annealing [32] and Gravitation search algorithm [39], Tabu Search [42] and Interior Search [43]. The ideas for swarm-intelligence based algorithms come from behavior of animals in the natures. The representatives of this type of algorithms are Ant Colony Optimization [16], Bee Colony Optimization [29], Bat algorithm [46], Firefly algorithm [47], Particle Swarm Optimization [31], Gray Wolf algorithm [38] and so on.

Between the best methods for solving combinatorial optimization problems is Ant Colony Optimization (ACO). The impulse for this method comes from the behavior of real ants. They always find the shortest path from the food to the nest. The ants leave a trail called a pheromone and follow the trail with the most concentrated pheromone.

The problem is represented with a help of a graph and the solutions are paths in a graph. The optimal solution for minimization problems is a shortest path, and for maximization problems is a longest path in a graph. The solution construction starts from random node of the graph and next nodes are included applying probabilistic rule. The pheromone is imitated by numerical information corresponding to the quality of the solution.

ACO is applied to many types of optimization problems. The idea for application of ant behavior for solving combinatorial optimization problems is done by Marco Dorigo twenty five years ago [15], [16], [18]. At the beginning it is applied on traveling salesman problem. Later it is successfully applied on a lot of complex optimization problems. During the years, various variants of ACO methodology was proposed: ant system [18]; elitist ants [18]; ant colony system [17]; max-min ant system [44]; rank-based ant system [18]; ant algorithm with additional reinforcement [21]. They differ in pheromone updating. For some of them is proven that they converge to the global optimum [18]. Fidanova et all [22]–[24] proposed semi-random start of the ants comparing several start strategies. The method can be adapted to dynamic changes of the problem in some complex biological problems [19], [20], [44].

Sometimes the metaheuristic algorithm, can not avoid local optimums. Appropriate Local Search (LS) procedure can help to escape them and to improve algorithm efficiency. We apply ACO on Multiple Knapsack Problem (MKP). A local search procedure, related with a specificity of MKP is constructed and combined with ACO to improve the algorithm performance and to avoid local optimums [24]. InterCriteria analysis (ICrA) is applied on the numerical results obtained by the traditional ACO and hybrid ACO in order to estimate the algorithms behavior. The approach ICrA has been applied for a large area of problems, e.g. [1], [2], [14], [25]. Published results show the applicability of the ICrA and the correctness of the approach.

The rest of the paper is organized as follows: The definition of the MKP is in Section 2. ACO algorithm is presented in Section 3. Local Search procedure is described in Section 4. Short notes on ICrA approach are presented in Section 5. Numerical results and a discussion are in Section 6. Conclusion remarks are done in Section 7.

II. Multiple Knapsack Problem

In knapsack problem is given a set of items with fixed weights and values. The aim is to maximize the sum of the values of the items in the knapsack, while remaining within the capacity of the knapsack. Each item can be selected only ones.

Multiple Knapsack Problem (MKP) is a generalization of the single knapsack problem and instead to have only one knapsack, there are many knapsacks with diverse capacity. Each item is assigned to maximum one of the knapsacks without violating any of the knapsacks capacity. The purpose is to maximize the total profit of the items in the knapsacks.

MKP is a special case of the generalized assignment problem [36]. It is a representative of the subset problems. Economical, industrial and other types of problems can be represented by MKP. Resource allocation in distributed systems, capital budgeting, cargo loading and cutting stock problems [30] are some of the applications of the problem. One important real problem which is represented as MKP is patients scheduling [3]. MKP is related with bin packing problem where the size of the bins can be variable [40] and cutting stock problem for cut row materials [30]. Other application is multi-processor scheduling on uniformly related machines [35]. Other difficult problem which leads to MKP is crypto-systems and generating keys [30]. One early application of MKP is tests generation [27]. MKP is a model large set of binary problems with integer coefficients [33], [36].

MKP is NP-hard problem and normally is solved with some metaheuristic method such as genetic algorithm [37], tabue search [48], swarm intelligence [34], ACO algorithm [21], [26].

We will define MKP as resource allocation problem, where m is the number of resources (the knapsacks) and n is the number of the objects. The object j has a profit p_j. Each resource has its own budget (knapsack capacity) and consumption r_{ij} of resource j by object i. The purpose is maximization of the profit within the limited budget.

The mathematical formulation of MKP can be as follows:

$$\max \textstyle\sum_{j=1}^{n} p_j x_j$$

$$\text{subject to} \textstyle\sum_{j=1}^{n} r_{ij} x_j \le c_i \quad i = 1, \dots, m \tag{1}$$

$$x_j \in \{0, 1\} \quad j = 1, \dots, n$$

There are m constraints in this problem, so MKP is also called m-dimensional knapsack problem. Let $I = \{1, \dots, m\}$ and $J = \{1, \dots, n\}$, with $c_i \ge 0$ for all $i \in I$. A well-stated MKP assumes that $p_j > 0$ and $r_{ij} \le c_i \le \sum_{j=1}^{n} r_{ij}$ for all $i \in I$ and $j \in J$. Note that the $[r_{ij}]_{m \times n}$ matrix and $[c_i]_m$ vector are both non-negative.

The MKP partial solution is represented by $S = \{i_1, i_2, \dots, i_j\}$ and the last element included to S, i_j is not used in the selection process for the next element. Thus the solution of MKP have not fixed length.

III. Ant Colony Optimization Algorithm

NP-hard problems require the use of huge resources and therefore cannot be solved by exact or traditional numerical methods, especially when they are large scale. We apply metaheuristic method aiming to find approximate solution using reasonable resources [18], [26].

Firs Marco Dorigo applies ideas coming from ants behavior to solve complicate optimization problems 30 years ago [16]. Some modifications are proposed by him and by other authors for algorithm improvement. The modifications concern pheromone updating [18]. The algorithm is problem dependent. Very important is representation of the problem by a graph. Thus the solutions represent paths in the graph. The ants look for an optimal path, taking in to account problem constraints.

The transition probability $P_{i,j}$, is a product of the heuristic information $\eta_{i,j}$ and the pheromone trail level $\tau_{i,j}$ related to the selection of node j if the previous selected node is i, where $i, j = 1, \dots., n$.

$$P_{i,j} = \frac{\tau_{i,j}^{a} \cdot \eta_{i,j}^{b}}{\sum\limits_{k \in Unused} \tau_{i,k}^{a} \cdot \eta_{i,k}^{b}}, \tag{2}$$

where $Unused$ is the set of unused nodes.

At the beginning the pheromone is initialized with a small constant value τ_0, $0 < \tau_0 < 1$. Every time the ants build a solution, the pheromone is bring up to date [18].The elements of the graph with more pheromone are more tempting to the ants.

The main update rule for the pheromone is:

$$\tau_{i,j} \leftarrow \rho \cdot \tau_{i,j} + \Delta\tau_{i,j}, \tag{3}$$

where parameter ρ decreases the value of the pheromone, like evaporation in a nature decreases the quantity of old pheromone. $\Delta\tau_{i,j}$ is a new deposited pheromone, which depends on the value of the objective function, corresponding to this solution.

The first step, when ACO is applied on some combinatorial optimization problem is representation of the problem by graph. In our case the items are related with the nodes of the graph and the edges fully connect the nodes. The pheromone is deposited on the arcs of the graph.

Second step is construction of appropriate heuristic information. This step is very important, because the heuristic information is the main part of the transition probability function and the search process depends mainly on it. Normally the heuristic information is a combination of problem parameters.

Let $s_j = \sum_{i=1}^{m} r_{ij}$. For heuristic information we use:

$$\eta_{ij} = \begin{cases} p_j^{d_1}/s_j^{d_2} & \text{if } s_j \neq 0 \\ p_j^{d_1} & \text{if } s_j = 0 \end{cases} \tag{4}$$

where $d_1 > 0$ and $d_2 > 0$ are parameters. Hence the objects with greater profit and less average expenses will be more desirable. Thus is increased the probability to include more items and most profitable items. This can lead to maximization of the total profit, which is the objective of this problem.

IV. Local Search Procedure

At times is used hybridization of the used method, for algorithm performance improvement. The goal is avoid some disadvantages of the main method. A possibility for hybridization is one of the methods to be basic and the other only helps to improve the solutions. Most used hybridization manner is local improvement or at the end of the iteration to apply some problem dependent local search procedure.

The Local Search (LS) procedure is used to perturbs current solution and to generate neighbor solutions [41]. LS generates neighbor solutions in a local set of neighbors. The best solution from the set is compared with the current solution. If it is better, it is accepted as a new current solution.

A LS procedure which is consistent with MKP has been developed and combined with ACO algorithm in our previous work [26]. The MKP solution is represented by binary string where 0 corresponds to not chosen item and 1 corresponds to item included in the solution. Two positions are randomly chosen. If the value of one of the positions is 0 we replace it with 1 and if the value of other position is 1 we replace it with 0 and vice versa. The feasibility of the new solution is verified. If the solution is feasible we compare it with the current (original) solution. The perturbed solution is accepted if its value of the objective function is greater, than of the original one.

We apply this LS procedure ones on each iteration on each solution, disregarding if the new constructed solution is better than current one or not. Thus the proposed LS works without significant increase of the used computational resources.

V. InterCriteria Analysis

Based on the apparatuses of index matrices [4], [6], [8], [9] and intuitionistic fuzzy sets (IFSs) [5], [7], [10], authors in [11] propose a new approach named InterCriteria analysis. Briefly presented, an intuitionistic fuzzy pair (IFP) [12] is an ordered pair of real non-negative numbers $\langle a, b \rangle$, where $a, b \in [0, 1]$ and $a + b \leq 1$, that is used as an evaluation of some object or process. According to [12], the components (a and b) of IFP might be interpreted as degrees of "membership" and "non-membership" to a given set, degrees of "agreement" and "disagreement", etc.

Let O denotes the set of all objects being evaluated, and $C(O)$ is the set of values assigned by a given criteria C (i.e., $C = C_p$ for some fixed p) to the objects, i.e.,

$$O \overset{\text{def}}{=} \{O_1, O_2, O_3, \ldots, O_n\},$$
$$C(O) \overset{\text{def}}{=} \{C(O_1), C(O_2), C(O_3), \ldots, C(O_n)\}.$$

Let $x_i = C(O_i)$. Then the following set can be defined:

$$C^*(O) \overset{\text{def}}{=} \{\langle x_i, x_j \rangle | i \neq j \,\&\, \langle x_i, x_j \rangle \in C(O) \times C(O)\}.$$

Further, if $x = C(O_i)$ and $y = C(O_j)$, $x \prec y$ if $i < j$ will be written.

In order to find the agreement of different criteria, the vectors of all internal comparisons for each criterion are constructed, which elements fulfill one of the three relations R, $\overline{R}$ and $\tilde{R}$. The nature of the relations is chosen such that for a fixed criterion C and any ordered pair $\langle x, y \rangle \in C^*(O)$:

$$\langle x, y \rangle \in R \Leftrightarrow \langle y, x \rangle \in \overline{R}, \tag{5}$$
$$\langle x, y \rangle \in \tilde{R} \Leftrightarrow \langle x, y \rangle \notin (R \cup \overline{R}), \tag{6}$$
$$R \cup \overline{R} \cup \tilde{R} = C^*(O). \tag{7}$$

For example, if "R" is the relation "<", then $\overline{R}$ is the relation ">", and vice versa.

When comparing two criteria the degree of "agreement"is determined as the number of matching components of the respective vectors (divided by the length of the vector for normalization purposes).

Let the respective degrees of "agreement" and "disagreement" are denoted by $\mu_{C,C'}$ and $\nu_{C,C'}$. In the most of the obtained pairs $\langle \mu_{C,C'}, \nu_{C,C'} \rangle$, the sum $\mu_{C,C'} + \nu_{C,C'}$ is equal to 1. However, there may be some pairs, for which this sum is less than 1. The difference

$$\pi_{C,C'} = 1 - \mu_{C,C'} - \nu_{C,C'} \tag{8}$$

is considered as a degree of "uncertainty.

VI. Computational Results and Discussion

The proposed hybrid ACO algorithm for MKP is tested on 10 test MKP instances from Operational Research Library "OR-Library" available within WWW access at *http://people.brunel.ac.uk/ mastjjb/jeb/info.html.* Every test problem consists of 100 items and 10 constraints/knapsacks. We prepare a software, which realizes our hybrid algorithm. The software is coded in $C{+}{+}$ program language and is run on Pentium desktop computer at 2.8 GHz with 4 GB of memory. The ACO algorithm parameters are fixed experimentally as follows:

- Number of iterations = 300, Number of ants = 20;
- ρ = 0.5, τ_0 = 0.5;
- a = 1, b = 1 and d_1 = 1.

We perform 30 independent runs with every one of the test instances, because the algorithm is stochastic and to guarantee the robustness of the average results. We apply ANOVA test for statistical analysis and thus we guarantee the significance of the difference between the average results. The names of

TABLE I: Test instances

Instance	Name	
	Hybrid ACO	Traditional ACO
MKP 100 × 10-01	$P1_h$	$P1_t$
MKP 100 × 10-02	$P2_h$	$P2_t$
MKP 100 × 10-03	$P3_h$	$P3_t$
MKP 100 × 10-04	$P4_h$	$P4_t$
MKP 100 × 10-05	$P5_h$	$P5_t$
MKP 100 × 10-06	$P6_h$	$P6_t$
MKP 100 × 10-07	$P7_h$	$P7_t$
MKP 100 × 10-08	$P8_h$	$P8_t$
MKP 100 × 10-09	$P9_h$	$P9_t$
MKP 100 × 10-10	$P10_h$	$P10_t$

TABLE II: Traditional ACO performance

	$P1_t$	$P2_t$	$P3_t$	$P4_t$	$P5_t$	$P6_t$	$P7_t$	$P8_t$	$P9_t$	$P10_t$
run1	22089	22452	20936	21481	21751	21810	21537	21634	22213	40594
run2	21954	22055	20966	21318	21606	21864	21659	21596	22398	40701
run3	21935	21912	21023	21318	21463	21912	21526	21516	22065	40647
run4	22030	21914	20732	21556	21519	21903	21470	21337	22191	40617
run5	21875	21990	21120	21451	21903	21970	21360	21689	22152	40489
run6	21970	21999	21114	21619	21736	21756	21426	21729	22125	40646
run7	21974	21990	21085	21740	21641	21654	21522	21515	22398	40714
run8	22041	22120	21032	21918	21811	22053	21584	21550	22109	40550
run9	21893	21924	21187	21335	21716	21864	21587	21725	22398	40581
run10	21984	22104	20822	21719	21767	21864	21509	21550	22078	40594
run11	21787	21950	21042	21661	21673	22047	21595	22067	22398	40659
run12	21916	22675	21182	21629	21818	21834	21426	21550	22101	40646
run13	22031	22027	20869	21490	21843	22123	21466	21508	22398	40584
run14	22188	21975	21203	21736	21811	21864	21601	21550	22398	40627
run15	21889	22119	21204	21740	21716	21824	21509	21573	22086	40515
run16	22009	22101	20877	21705	21952	21779	21409	21506	22039	40498
run17	21880	21990	21085	21531	21736	21713	21394	21579	22398	40680
run18	21958	22102	20872	21335	21811	22053	21596	21550	22105	40589
run19	22015	21963	20841	21861	21581	21864	21624	21729	22324	40404
run20	22054	22027	21007	21815	21679	22140	21392	21729	22398	40367
run21	22093	22027	20833	21607	21811	21816	21509	21496	22039	40496
run22	22074	22065	20960	21701	21711	21903	21509	21496	22398	40737
run23	22003	22027	20976	21759	21685	21864	21522	21339	22039	40594
run24	22169	22005	21003	21490	21735	21898	21511	21520	22156	40317
run25	22091	22106	20808	21964	21622	21840	21434	21614	22398	40664
run26	21945	21899	21103	21335	21417	22053	21426	21516	22059	40319
run27	22086	22196	21069	21774	21944	22241	21590	21629	22398	40728
run28	21926	21975	21003	21437	21922	21864	21531	21503	22398	40636
run29	21929	22177	20799	21681	21736	21907	21479	21530	22398	40498
run30	22030	21931	20925	22061	21434	21864	21509	21366	22398	40756

the test instances are presented in Table I. In Table II and Table III observed numerical results for all 30 runs are listed.

In Table IV are reported average results for every one of the test instances over 30 runs. We compare ACO algorithm combined with local search procedure (hybrid ACO) with traditional ACO algorithm. On the last row is reported average computational time, in seconds, of the two variants of ACO algorithm.

Table IV shows that for eight of ten instances hybrid ACO algorithm outperforms the traditional one. For the instances MKP 100 × 10-02 and MKP 100 × 10-10 the results are statistically the same. The main problem with hybrid algo-

TABLE III: Hybrid ACO performance

	$P1_h$	$P2_h$	$P3_h$	$P4_h$	$P5_h$	$P6_h$	$P7_h$	$P8_h$	$P9_h$	$P10_h$
run1	22206	22047	21292	21885	21811	21940	21509	22004	22270	40647
run2	21968	22186	21089	21962	21811	21957	21509	21616	22097	40557
run3	21970	22074	21233	22139	21716	21934	21530	21550	22087	40598
run4	22130	22028	21687	21701	21811	22024	21415	21729	22294	40679
run5	22107	22168	21020	21885	21811	22047	21522	21729	22285	40647
run6	22138	22027	21222	21736	21798	21980	21522	21551	22257	40522
run7	22367	22027	21416	21962	21811	21900	21437	21729	22140	40538
run8	22051	22028	21261	21420	21790	22048	21509	21648	22125	40710
run9	21867	22104	20861	21780	21811	21985	21531	21729	22398	40710
run10	21910	22027	21090	21666	21811	21987	21537	21729	22398	40647
run11	22133	22027	20758	21854	21844	22011	21655	21729	22398	40583
run12	22164	22074	20848	21885	21798	22053	21330	21653	22154	40662
run13	21949	22044	21090	22099	21736	21987	21409	21729	22398	40664
run14	22067	22102	20954	21921	21798	21987	21587	21729	22117	40683
run15	21889	22027	21236	21801	21811	22113	21509	21729	22069	40689
run16	21999	22213	21185	21561	21811	21900	21509	21650	22429	40636
run17	21926	22065	21017	22174	21914	21937	21509	21550	22191	40489
run18	21914	22027	21097	21542	21855	21864	21509	21550	22247	40714
run19	22008	22028	20953	21490	21796	22053	21624	21683	22398	40677
run20	21840	22151	21166	21893	21804	21987	21418	21729	22193	40728
run21	21993	22155	20839	21656	21811	21891	21584	21550	22398	40565
run22	22130	22106	21134	21962	21798	22063	21533	21658	22479	40742
run23	21958	22060	20881	21885	21811	21987	21426	21729	22152	40503
run24	22014	22027	21373	22023	21811	21924	21511	21729	22152	40514
run25	22072	22027	20925	21953	21811	21987	21509	21697	22123	40650
run26	22088	22104	20857	22052	21974	21987	21509	21550	22429	40751
run27	21928	22027	20934	21864	21811	21987	21509	21550	22218	40658
run28	21933	22110	20916	21885	21811	22121	21509	21689	22398	40499
run29	21970	22027	21281	21793	21832	21987	21509	21729	22398	40598
run30	21993	22027	21064	21951	21811	22050	21509	21550	22479	40540

TABLE IV: Comparison of ACO performance

Instance	Hybrid ACO	Traditional ACO
MKP 100 × 10-01	**22022.73**	21989.43
MKP 100 × 10-02	22071.46	22081.36
MKP 100 × 10-03	**21089.3**	21027.63
MKP 100 × 10-04	**21846**	21635.3
MKP 100 × 10-05	**21814.3**	21717.3
MKP 100 × 10-06	**21989.26**	21869.73
MKP 100 × 10-07	**21506.26**	21477.3
MKP 100 × 10-08	**21672.53**	21606.43
MKP 100 × 10-09	**22272.36**	22257
MKP 100 × 10-10	40626.66	40623.26
computational time	64.052 s	65.552 s

rithms, when some global method is combined with local search procedure, is increasing of computational time. We try to propose efficient and in a same time less time consuming local search. We only change randomly chosen position in a solution to 0, if it is 1 and another randomly chosen position to 1 if it is 0. Thus is generated only one neighbor solution. If this solution is better than the current one, it is accepted and used for pheromone updating instead of the solution constructed by the ant. We apply this procedure to each of the solutions. As is seen from Table IV the increase of computational time, when our local search is applied is only 2.34%.

Thus we can conclude that proposed local search procedure is efficient and effective. The algorithm performance is improved, without significant increase of the computational time.

To support these claims, the obtained numerical results were analyzed using ICrA. The input matrix for ICrA has the following form index matrix Table V:

The obtained by ICrA results are listed in Table VI (μ-values) and Table VII (ν-values). The π-values are also presented (see Table VIII). The results between the same instances but for different ACO algorithms are presented. For example, relations between $P1_t - P1_h$, $P2_t - P2_h$, $P3_t - P3_h$, etc. are considered for further analysis (presented in bold results in Tables VI, VII and VIII).

According to [13] the results show that the considered

TABLE V: Index matrix for ICrA

$$\begin{array}{c|cccc}
 & \text{run1} & \text{run2} & \dots & \text{run30} \\ \hline
P1_t & val_{P1_{t,1}} & val_{P1_{t,2}} & \vdots & val_{P1_{t,30}} \\
P2_t & val_{P2_{t,1}} & val_{P2_{t,2}} & \vdots & val_{P2_{t,30}} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
P10_t & val_{P10_{t,1}} & val_{P10_{t,2}} & \vdots & val_{P10_{t,30}} \\
P1_h & val_{P1_{h,1}} & val_{P1_{h,2}} & \vdots & val_{P1_{h,30}} \\
P2_h & val_{P2_{h,1}} & val_{P2_{h,2}} & \vdots & val_{P2_{h,30}} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
P10_h & val_{P10_{h,1}} & val_{P10_{h,2}} & \vdots & val_{P10_{h,30}}
\end{array} \tag{9}$$

TABLE VI: Degree of agreement – $\mu_{C,C'}$-values

μ	$P1_t$	$P2_t$	$P3_t$	$P4_t$	$P5_t$	$P6_t$	$P7_t$	$P8_t$	$P9_t$	$P10_t$
$P1_h$	**0.55**	0.46	0.51	0.5	0.45	0.38	0.45	0.46	0.37	0.55
$P2_h$	0.42	**0.37**	0.48	0.3	0.48	0.39	0.36	0.37	0.34	0.36
$P3_h$	0.57	0.57	**0.44**	0.5	0.46	0.46	0.46	0.43	0.39	0.47
$P4_h$	0.47	0.39	0.55	**0.4**	0.35	0.47	0.4	0.41	0.51	0.61
$P5_h$	0.25	0.34	0.33	0.27	**0.32**	0.36	0.35	0.34	0.34	0.34
$P6_h$	0.4	0.45	0.5	0.57	0.43	**0.5**	0.44	0.46	0.46	0.48
$P7_h$	0.41	0.3	0.44	0.38	0.4	0.38	**0.52**	0.51	0.36	0.38
$P8_h$	0.4	0.42	0.38	0.42	0.4	0.37	0.39	**0.42**	0.42	0.29
$P9_h$	0.45	0.38	0.35	0.44	0.49	0.49	0.39	0.39	**0.42**	0.41
$P10_h$	0.55	0.52	0.5	0.52	0.46	0.63	0.48	0.54	0.38	**0.38**

TABLE VII: Degree of disagreement – $\nu_{C,C'}$-values

ν	$P1_t$	$P2_t$	$P3_t$	$P4_t$	$P5_t$	$P6_t$	$P7_t$	$P8_t$	$P9_t$	$P10_t$
$P1_h$	**0.44**	0.51	0.47	0.48	0.52	0.54	0.52	0.5	0.4	0.43
$P2_h$	0.44	**0.46**	0.38	0.56	0.37	0.4	0.48	0.46	0.38	0.49
$P3_h$	0.43	0.4	**0.55**	0.49	0.52	0.47	0.5	0.54	0.4	0.51
$P4_h$	0.49	0.55	0.42	**0.55**	0.59	0.43	0.54	0.53	0.26	0.36
$P5_h$	0.46	0.36	0.38	0.43	**0.37**	0.35	0.37	0.34	0.27	0.37
$P6_h$	0.51	0.45	0.41	0.33	0.46	**0.36**	0.44	0.42	0.31	0.43
$P7_h$	0.41	0.5	0.37	0.43	0.41	0.4	**0.27**	0.28	0.34	0.43
$P8_h$	0.37	0.35	0.39	0.35	0.36	0.36	0.37	**0.33**	0.24	0.47
$P9_h$	0.47	0.53	0.57	0.48	0.41	0.4	0.52	0.51	**0.34**	0.51
$P10_h$	0.43	0.44	0.48	0.45	0.5	0.29	0.47	0.42	0.39	**0.6**

criteria pairs, are in dissonance or in strong dissonance. This means that the both compared ACO algorithms (hybrid and traditional ones) performed differently in case of all 10 various instances.

The results obtained form ICrA are correct and reliable, taking into account the observed values of $\pi_{C,C'}$-values. Only for relations between $P5_t - P5_h$, $P8_t - P8_h$ and $P9_t - P9_h$, there are some high $\pi_{C,C'}$-values, respectively 0.31, 0.25 and 0.24. The obtained estimates for the degree of agreement and the degree of disagreement have a high degree of uncertainty.

VII. Conclusion

In this paper we propose hybrid ACO algorithm for solving MKP. The algorithm is combination of traditional ACO algorithm and local search procedure. Proposed algorithm is tested on 10 benchmark MKP. The achieved results show the efficiency and effectiveness of the proposed local search procedure. The hybrid algorithm performs better than the traditional one, while the calculation time increases only with 2.34%.

Obtained results are analyzed by ICrA approach. The anal-

TABLE VIII: Degree of uncertainty – $\pi_{C,C'}$-values

π	$P1_t$	$P2_t$	$P3_t$	$P4_t$	$P5_t$	$P6_t$	$P7_t$	$P8_t$	$P9_t$	$P10_t$
$P1_h$	**0.01**	0.03	0.02	0.02	0.03	0.08	0.03	0.04	0.23	0.02
$P2_h$	0.14	**0.17**	0.14	0.14	0.15	0.21	0.16	0.17	0.28	0.15
$P3_h$	0	0.03	**0.01**	0.01	0.02	0.07	0.04	0.03	0.21	0.02
$P4_h$	0.04	0.06	0.03	**0.05**	0.06	0.1	0.06	0.06	0.23	0.03
$P5_h$	0.29	0.3	0.29	0.3	**0.31**	0.29	0.28	0.32	0.39	0.29
$P6_h$	0.09	0.1	0.09	0.1	0.11	**0.14**	0.12	0.12	0.23	0.09
$P7_h$	0.18	0.2	0.19	0.19	0.19	0.22	**0.21**	0.21	0.3	0.19
$P8_h$	0.23	0.23	0.23	0.23	0.24	0.27	0.24	**0.25**	0.34	0.24
$P9_h$	0.08	0.09	0.08	0.08	0.1	0.11	0.09	0.1	**0.24**	0.08
$P10_h$	0.02	0.04	0.02	0.03	0.04	0.08	0.05	0.04	0.23	**0.02**

ysis shows that the both algorithms performs differently for the considered 10 instances, i.e. the behavior of the proposed hybrid ACO is importantly different from that of the traditional ACO algorithm, or the local search procedure perturbs significantly the search process.

Through the application of ICrA approach the efficiency and effectiveness of the proposed hybrid ACO agorithm are confirmed.

Acknowledgment

The development of the proposed hybrid ACO algorithm with local search procedure has been funded by the Grant DFNI DN 12/5 and Grant No BG05M2OP001-1.001-0003, financed by the Science and Education for Smart Growth Operational Program and co-financed by the European Union through the European structural and Investment funds. The study of ACO algorithms behavior based on ICrA approach has been funded by the Grant KP-06-N22/1.

Author Contributions: The authors contributed equally to the work.

References

[1] M. Angelova, O. Roeva and T. Pencheva. InterCriteria Analysis of Crossover and Mutation Rates Relations in Simple Genetic Algorithm. In *Proceedings of the 2015 Federated Conference on Computer Science and Information Systems*, Vol. 5, pages 419–424, 2015.

[2] Antonov A., Dependencies Between Model Indicators of the Basic and the Specialized Speed in Hockey Players Aged 13-14, Trakia Journal of Sciences, Vol. 18, Suppl. 1, pp 647-657, 2020.

[3] Idil Arsik, Pinar Keskinocak, Jennifer Coppola, Kirthana Hampapur, Yitong He, Haozheng Jiang,Dani Regala, Nick Tailhardat, and Kristin Goin. Effective and equitable appointment scheduling in rehabilitation centers.INFORMS Annual Meeting 2017, Oct. 22-25 2017.

[4] K. Atanassov. *Index matrices: Towards an augmented matrix calculus*. Springer International Publishing Switzerland, 2014.

[5] K. Atanassov. Intuitionistic Fuzzy Sets. VII ITKR Session, Sofia, 20-23 June 1983, Reprinted: *International Journal Bioautomation*, 20(S1):S1–S6, 2016.

[6] K. Atanassov. Generalized index matrices. *Comptes rendus de l'Academie bulgare des Sciences*, 40(11):15–18, 1987.

[7] K. Atanassov. *On intuitionistic fuzzy sets theory*. Springer, Berlin, 2012.

[8] K. Atanassov. On index matrices, Part 1: Standard cases. *Advanced Studies in Contemporary Mathematics*, 20(2):291–302, 2010).

[9] K. Atanassov. On index matrices, Part 2: Intuitionistic fuzzy case. *Proceedings of the Jangjeon Mathematical Society*, 13(2):121–126, 2010.

[10] K. Atanassov. Review and New Results on Intuitionistic Fuzzy Sets, Mathematical Foundations of Artificial Intelligence Seminar, Sofia, 1988, Preprint IM-MFAIS-1-88. Reprinted: *International Journal Bioautomation*, 20(S1):S7–S16, 2016.

[11] K. Atanassov, D. Mavrov and V. Atanassova. Intercriteria decision making: A new approach for multicriteria decision making, based on index matrices and intuitionistic fuzzy sets. *Issues in on Intuitionistic Fuzzy Sets and Generalized Nets*, 11:1–8, 2014.

[12] K. Atanassov, E. Szmidt and J. Kacprzyk. On intuitionistic fuzzy pairs. *Notes on Intuitionistic Fuzzy Sets*, 19(3):1–13, 2013.

[13] K. Atanassov, V. Atanassova and G. Gluhchev. InterCriteria Analysis: ideas and problems. *Notes on Intuitionistic Fuzzy Sets*, 21(1):81–88, 2015.

[14] V. Atanassova, D. Mavrov, L. Doukovska and K. Atanassov. Discussion on the Threshold Values in the InterCriteria Decision Making Approach. *Notes on Intuitionistic Fuzzy Sets*, 20(2): 94–99, 2014.

[15] M. Birattari, T. Stutzle, L.Paquete, K. Varrentrapp, A racing algorithm for configuring metaheuristics, Proceedings of the Genetic and Evolutionary Computation Conference, pp. 11–18, 2002.

[16] Bonabeau E., Dorigo M. and Theraulaz G., *Swarm Intelligence: From Natural to Artificial Systems*, New York,Oxford University Press, 1999.

[17] M. Dorigo, L. Gambardella, Ant colony system : A cooperative learning approach to the traveling salesman problem, IEEE Transactions on Evolutionary Computation vol. 1, 53–66, 1996.

[18] Dorigo M, Stutzle T., *Ant Colony Optimization*, MIT Press, 2004.

[19] S. Fidanova, I. Lirkov, 3d protein structure prediction, Analele Universitatii de Vest Timisoara, vol. XLVII, pp. 33–46, 2009.

[20] S. Fidanova, An improvement of the grid-based hydrophobic-hydrophilic model, Int. J. Bioautomation, vol. 14, pp. 147–156, 2010.

[21] S. Fidanova, ACO algorithm with additional reinforcement, Int Conf. from Ant Colonies to Artificial Ants, Lecture Notes in Computer Science 2463, 292–293, 2003.

[22] S. Fidanova,K. Atanassov, P. Marinov, Generalized nets and ant colony optimization, Bulg. Academy of Sciences Pub. House, 2011.

[23] S. Fidanova, K. Atanassov, P. Marinov, Start strategies of ACO applied on subset problems, Numerical Methods and Applications, Lecture Notes in Computer Science, 6046, 248–255, 2011.

[24] S.Fidanova, K. Atanassov, P. Marinov, Intuitionistic fuzzy estimation of the ant colony optimization starting points, Large Scale Scientific Computing, Lecture Notes in Computer Science 7116, 219–226, 2012.

[25] S. Fidanova, O. Roeva and M. Paprzycki. InterCriteria Analysis of ACO Start Strategies, In *Proceedings of the 2016 Federated Conference on Computer Science and Information Systems*, Vol. 8, pages 547–550, 2016.

[26] Fidanova S. Hybrid Ant Colony Optimization Algorithm for Multiple Knapsack Problem. 5th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE), IEEE, 2021, DOI:10.1109/ICRAIE51050.2020.9358351, 1-5.

[27] Feuerman, Martin; Weiss, Harvey (April 1973). "A Mathematical Programming Model for Test Construction and Scoring". Management Science. 19 (8): 961–966

[28] D.E. Goldberg, B. Korb, K. Deb, Messy Genetic Algorithms: Motivation Analysis and First Results, Complex Systems, vol. 5(3), pp. 493 – 530, 1989.

[29] D. Karaboga, B. Basturk, Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems, Advances in Soft Computing: Foundations of Fuzzy Logic and Soft Computing, LNCS 4529, 789-798, 2007.

[30] Kellerer H., Pferschy U., Pisinger D. (2004) Multiple Knapsack Problems. In: Knapsack Problems. Springer, Berlin,

[31] J. Kennedy, R. Eberhart, Particle Swarm Optimization. Proceedings of IEEE International Conference on Neural Networks. IV. pp. 1942–1948, 1995.

[32] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing Science (New York, N.Y.) (New York, N.Y.), 13 (220) (1983), pp. 671-680

[33] G.Kochenberger, G.McCarl, F.Wymann, An heuristic for general integer programming, Decision Sciences vol. 5, 34–44, 1974.

[34] Jonas Krause, Jelson Cordeiro, Rafael Stubs Parpinelli, and Heitor Silverio Lopes. A survey ofswarm algorithms applied to discrete optimization problems. InSwarm Intelligence and Bio-Inspired Computation, pages 169–191. Elsevier, 2013

[35] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys. Sequencing and scheduling:algorithms and complexity. In S. C. Graves et al. , editor,Handbooks in OR and MS, volume 4, pages445–522. Elsevier Science Publishers, 1993

[36] G.Leguizamon, Z.Michalevich, A new version of ant system for subset problems, Int. Conf. on Evolutionary Computations vol. 2, 1459–1464, 1999.

[37] Qing Liu, Tomohiro Odaka, Jousuke Kuroiwa, Haruhiko Shirai, and Hisakazu Ogura. A newartificial fish swarm algorithm for the multiple knapsack problem.IEICE TRANSACTIONS onInformation and Systems, 97(3):455–468, 2014

[38] S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey Wolf Optimizer, Advances in Engineering Software, vol. 69, pp. 46–61, 2014.

[39] M.R. Mosavi, M. Khishe, G.R. Parvizi, M.J. Naseri, M. Ayat, Training multi-layer perceptron utilizing adaptive best-mass gravitational search algorithm to classify sonar dataset Archive of Acoustics, 44 (1) (2019), pp. 137-151

[40] F. D. Murgolo. An efficient approximation scheme for variable-sized bin packing.SIAM Journal onComputing, 16(1):149–161, 1987.

[41] Schaffer, A. A., Yannakakis, M.: *Simple Local Search Problems that are Hard to Solve*. Society for Industrial Applied Mathematics Journal on Computing, Vol 20 (1991) 56–87.

[42] I.H. Osman, Metastrategy simulated annealing and tabue search algorithms for the vehicle routing problem, Annals of Operations Research, 41 (4) (1993), pp. 421-451

[43] S. Ravakhah, M. Khishe, M. Aghababaee, E. Hashemzadeh, Sonar false alarm rate suppression using classification methods based on interior search algorithm, International Journal of Computer Science and Network Security, 17 (7) (2017), pp. 58-65

[44] T. Stutzle, H. Hoos, Max min ant system, Future Generation Computer Systems, vol. 16, 889–914, 2000.

[45] P.A. Vikhar, Evolutionary algorithms: A critical review and its future prospects, Proceedings of the 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC). Jalgaon, pp. 261–265, 2016.

[46] X.S. Yang, A New Metaheuristic Bat-Inspired Algorithm, Nature Inspired Cooperative Strategies for Optimization, Studies in Computational Intelligence, 284: 65–74, 2010.

[47] X.S. Yang, Nature-Inspired Metaheuristic Algorithms, Luniver Press, 2008.

[48] Andrew J Woodcock and John M Wilson. A hybrid tabue search/branch and bound approach tosolving the generalized assignment problem.European journal of operational research, 207(2):566–578, 2010

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 181–189
DOI: 10.15439/2021F45
ISSN 2300-5963 ACSIS, Vol. 25

On the Representation of Human Motions and Distance-based Retargeting

Simon B. Hengeveld* A. Mucherino,*
*IRISA, University of Rennes 1, Rennes, France.
simon.hengeveld@irisa.fr, antonio.mucherino@irisa.fr

***Abstract*—Distance-based motion adaptation leads to the formulation of a dynamical Distance Geometry Problem (dynDGP) where the involved distances simultaneously represent the morphology of the animated character, as well as a possible motion. The explicit use of inter-joint distances allows us to easily verify the presence of joint contacts, which one generally wishes to preserve when adapting a given motion to characters having a different morphology. In this work, we focus our attention on suitable representations of human-like animated characters, and study the advantages (and disadvantages) in using some of them. In the initial works on distance-based motion adaptation, a $3n$-dimensional vector was employed for representing the positions of the n joints of the character at a given frame. Here, we investigate the use of another, very popular in computer graphics, representation that basically replaces every joint position in the three-dimensional space with a set of three sorted Euler angles. We show that the latter can in fact be useful for avoiding some of the artifacts that were observed in previous computational experiments, but we argue that this Euler-angle representation, from a motion adaptation point of view, does not seem to be the optimal one. By paying particular attention to the degrees of freedom of the studied representations, it turns out that a novel character representation, inspired by representations used in structural biology for molecules, may allow us to reduce the character degrees of freedom to their minimal value. As a result, statistical analysis on human motion databases, where the motions are given with this new representation, can potentially provide important insights on human motions. This study is an initial step towards the identification of a full set of constraints capable of ensuring that unnatural postures for humans cannot be created while tackling motion adaptation problems.**

I. Introduction

MOTION adaptation is a fundamental problem arising in computer graphics [6], [7], [8]. From a given motion for a given character, the interest lies in finding the way to *impose* the same (or a very "similar") motion to another character having a different morphology. The implications that a robust solution to this problem can give in the context of computer graphics are evident and include, for example, the partial automation in the production of animated pictures, as well as the conception and development of computer games. In the context of computer graphics, this problem is also known as *motion retargeting*.

We consider a simple undirected graph $G = (V, E)$ to represent the skeletal anatomy of our characters to be animated. The vertex set V of the graph contains the *joints* of the character, while edges in E represent rigid "bars" between some pairs of joints, that in this context are often referred to as character *bones*. In this work, we will focus our attention on graphs G that are trees, where every joint has one unique *parent* joint. In fact, a tree G is particularly suitable to represent the skeletal structure of human bodies. Together with G, the representation of our characters is generally integrated with the function

$$\chi : v \in V \longrightarrow \chi(v) \in \mathbb{R}^3,$$

which assigns a three-dimensional position, w.r.t. its own parent, to every joint of the character. We remark that the function χ not only encodes the *initial posture* of the character (i.e. the relative position of all character joints in absence of movement), but it also implicitly provides information about the *morphology* of the character. In fact, for every bone $\{u, v\} \in E$, where the joint u is parent of the joint v in the tree G, the real value $||\chi(v)||$ corresponds to the length of the bone $\{u, v\}$ (the symbol $|| \cdot ||$ represents the Euclidean norm). Notice that, in the context of graph rigidity [1], [9], the pair (G, χ) is also called a *skeletal structure*. An example of skeletal structure for a human character is given in Fig. 1.

Motion adaptation asks, given a motion and a character (G, χ), whether it is possible to adapt this motion so that a different character $(G, \hat{\chi})$, having the same skeletal anatomy G but a different morphology $\hat{\chi}$, can actually perform the same motion [6]. This is not a trivial problem, because even small changes in the morphology can make the original movement "look" different to the viewer. In particular, joint contacts play a very important role: joint contacts that are not preserved from the original motion are likely to give the viewer the impression that the motion is "different"; the same impression can be given by a motion performed by a morphological different character where new joint contacts, originally not appearing in the motion, are introduced. We point out that the concept of *contact* is closely related to the concept of *proximity* (or *distance*).

This work elaborates on some previous research on distance-based motion adaptation [3], [17]. The main idea is to represent the character animations by an alternative representation where the distance information is exploited. Inter-joint distances, in fact, can at the same time encode the morphology of the character (given by the function χ), as well as the relative movements of the joints that are not connected by an edge of G. This alternative representation seems to be very convenient to the purposes of motion adaptation, because it allows for an easy detection of joint contacts. This is done by the simple verification of the values of the corresponding distances (the contact detection is otherwise not trivial in other usually employed representations for the motions, the ones we will discuss in Section II).

Evolving inter-joint distances, however, do not allow for defining the most efficient representation of a character motion. Even when making the hypothesis that all involved distances are known at a high precision level, the set of evolving distances are likely to carry highly redundant information, and any modification on one single distance would imply the infeasibility of the distance set. Therefore, we use distance-based representations of the motions only for generating instances of the *dynamical* Distance Geometry Problem (dynDGP) [16] where the distance information is extracted, at the same time, from the original motion, as well as from the target morphology $\hat{\chi}$. The solution to the dynDGP instance will consist in a motion for the character $(G, \hat{\chi})$ where the error on the given distances is minimized. Naturally, some distances (such as the bone lengths indicated by the function $\hat{\chi}$) have a higher importance w.r.t. others.

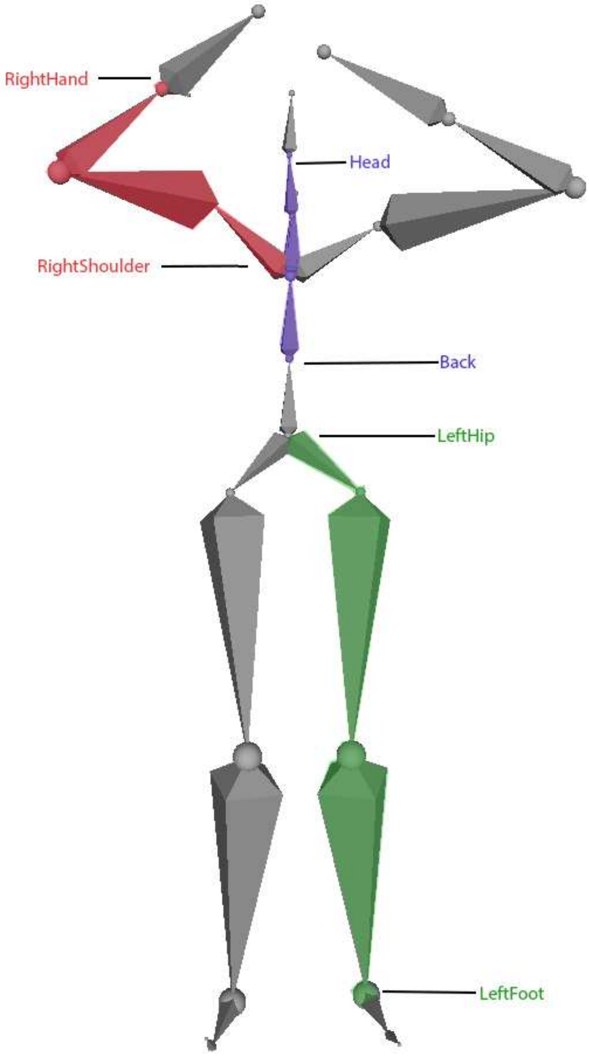


Fig. 1. An example of skeletal structure (G, χ). Marked in red, violet and green, the joints and bones involved in the computation of the vector and torsion angles for, respectively, the joint representing the RightHand, the Head and the LeftFoot of the human character.

An important part of this work is devoted to the representations employed for the resulting motions. The solution to the generated dynDGP instance allows us to convert the high-redundancy and sensitive distance-based representation, which is however able to efficiently control character inter-joint proximity, to another “more convenient” representation. In the initial works in [17], every joint was represented with its three-dimensional coordinates in the Euclidean space, but it was subsequently shown that the use of this representation can potentially introduce artifacts in the resulting motions [3].

In this work, we make two steps forward in the identification of an optimal representation for the character motions. First, in the experiments we will present on motion adaptation, we will use the typical (and very popular) representation used in the context of computer graphics [13], which is based on the rotations given by three Euler angles on every character bone. We will show that some of the previously observed artifacts disappear with the employment of this Euler-angle representation. Second, we will present a study on the degrees of freedom of different possible representations for the motions, which shows that the Euler-angle representation is not the optimal one in terms of degrees of freedom. In fact, this representation does not ensure that there is a bijective correspondence between character postures and variable values. Therefore, we propose another representation, inspired by a molecular representation that is widely used in the context of structural biology. By using this new representation, we performed a statistical analysis on a database of human character motions, with the final aim of deriving simple constraints capable to delimit the postures that are natural for a human character.

The rest of the paper is organized as follows. Section II will discuss different existing mathematical representations of character motions, while Section III will introduce our novel character representation inspired by the representation of biological molecules. By taking into consideration the very popular character representation which associates three Euler angles to each character joint, we will revisit in Section IV our distance-based motion adaptation approach, and some computational experiments will be presented in Section IV-A. By taking into consideration, instead, our representation of the motions introduced in Section III, we will present a statistical analysis on some of the variables used in this representation. Finally, Section VI will conclude this article with some directions for future works.

II. A MATHEMATICAL DESCRIPTION OF MOTIONS

In the wide computer graphics literature on motion retargeting, it is not common to find rigorous descriptions of the objects that come to play. An initial attempt can be found in [15]. In this section, we briefly re-propose the mathematical description given in the previous reference, and extend it to the purposes of our article.

Let $G = (V, E)$ be a simple undirected graph representing the skeletal *anatomy* of the characters. The pair (G, χ), coupling the graph G with the function $\chi : v \in V \to \chi(v) \in \mathbb{R}^3$, represents a skeletal structure, which completes the description of the character with its *morphology*.

We point out that the graph G generally contains a fictive root joint $v_0 \in V$ that is always associated to the origin (i.e. $\chi(v_0) = (0,0,0)$) of the three-dimensional Euclidean space where the motions take place. The joint v_0 is naturally not part of character; it is only used to encode the global orientation of the character in its environment. With this constraint on the root joint $v_0 \in V$, and by using the values $\chi(v)$ for all other joints $v \in V$, a special realization x of G can be immediately derived. Let $p : v \in V \setminus \{v_0\} \to p(v) \in V$ be the function that assigns the parent to each vertex of G (exception made for the root v_0). The realization x can therefore be constructed as follows:

$$x : v \in V \longrightarrow \begin{cases} (0,0,0) & \text{if } v = v_0, \\ \chi(p(v)) + \chi(v) & \text{otherwise.} \end{cases} \qquad (1)$$

This graph realization corresponds to the character posture in absence of any movement, and we will refer to this special posture as *the posture 0*.

One trivial representation of the motions is the one consisting in extending the function χ (or similarly, the function x) to the several time frames forming the motion (instead of using them for the definition of posture 0 only). This representation uses three variables per joint, and therefore it gives $3|V|$ degrees of freedom to the character. We point out, however, that this representation has a main drawback: when the values of $\chi(v)$ can vary over time for every vertex $v \in V$, the length $||\chi(v)||$ of the bones $\{u, v\} \in E$ are subject to change over time. We remark that this is an useless degree of freedom, that is likely to spoil the representations every time it is not made sure that all length values $||\chi(v)||$ remain constant during the motions.

A popular way to represent the motions consists instead in assigning the three Euler angles θ (pitch), ϕ (roll) and η (yaw) to every bone of the skeletal structure representing our character [5]. In this representation, the bone lengths $||\chi(v)||$ are constant by definition, while the three Euler angles are in charge to rotate the bones $\{u, v\} \in E$ in order to identify the position of the joint v w.r.t. the joint u (where u is the parent of v in G). In other words, the Euler-angle representation ensures that the character morphology χ remains constant during the motion.

Posture 0 corresponds in this representation to a list of Euler angles (the triplet of angles θ, ϕ and η for each bone in the skeletal structure) that are equal to 0. More specifically, the character motion can be encoded as a sequence over a predetermined number m of frames $t \in T$, with $T = \{1, 2, \dots, m\}$, of the Euler angles for every bone $\{u, v\} \in E$ forming the animated character:

$$\rho : (v,t) \in V \setminus \{v_0\} \times T \to \left(\theta^t_{p(v),v}, \phi^t_{p(v),v}, \eta^t_{p(v),v}\right) \in [0, 2\pi)^3.$$

We point out that the order for the three Euler angles is generally not fixed, and may be specific to the joint; for simplicity, we will suppose in the following that this order is constant: first $\theta^t_{p(v),v}$, then $\phi^t_{p(v),v}$, finally $\eta^t_{p(v),v}$. This order does not have any impact in our discussion. A more important remark is that, given ρ, the realization x in Equ. (1) can be simply derived [4].

Notice that the total number of bones in the character is $|V| - 1$, because G is a tree: the number of degrees of freedom for the two representations x and ρ is essentially the same. This simple fact immediately implies that ρ is not an optimal representation for the motions. Apart from some drawbacks already discussed in [4] about the singularity and the accuracy of this representation, we can remark that the set of Euler angles gives to the animated characters unnecessary degrees of freedom. This claim is supported by the following two observations. First of all, there always exist different triplets of Euler angles for a joint v that, when applied to the bone $\{p(v), v\}$, can place this joint in the same position [22]. Secondly, not all triplets of possible Euler angles correspond to natural human postures (imagine for example the postures where there are obtuse angles at knees or elbows). In [12], a specific representation was proposed for the human shoulders where scapulo-thoracic constraints and joint sinus cones are used to cope with unrealistic postures at the upper part of the body that a basic Euler-angle representation may give. In spite of these observed limitations, the function ρ has remained predominant in widely-used file formats for character motions [13].

III. A novel representation

In this section, a new character representation for which we are able to count a smaller number of degrees of freedom is presented. To this aim, we introduce the two new angles ζ_v and ω_v, that will sometimes replace the triplet of Euler angles employed in the function ρ for the representation of a joint. Differently from a Euler angle, which depends only on the joint v itself and on its parent $p(v)$, the *vector angle* ζ_v depends on v, $p(v)$, as well as on $(p \circ p)(v)$. Moreover, the *torsion angle* ω_v depends on v, $p(v)$, $(p \circ p)(v)$, as well as on $(p \circ p \circ p)(v)$.

The reader may have noticed that, in spite of this extra dependence, only the main joint v is indicated as a subscript of the angle names ζ_v and ω_v; this is done in order to have a lighter notation. However, it will be supposed that, every time these two angles are taken into account, the necessary ancestors of v ($p(v)$, $(p \circ p)(v)$, and finally $(p \circ p \circ p)(v)$ for ω_v, all exist.

Definition 1 *Given a skeletal structure (G, χ) and one realization x, the vector angle ζ_v for the joint v in this realization is the smallest angle formed by the line passing through $x((p \circ p)(v))$ and $x(p(v))$, and the line passing through $x(p(v))$ and $x(v)$.*

We can remark that variations on values of the vector angle ζ_v imply movements of the joint v. However, as for the Euler

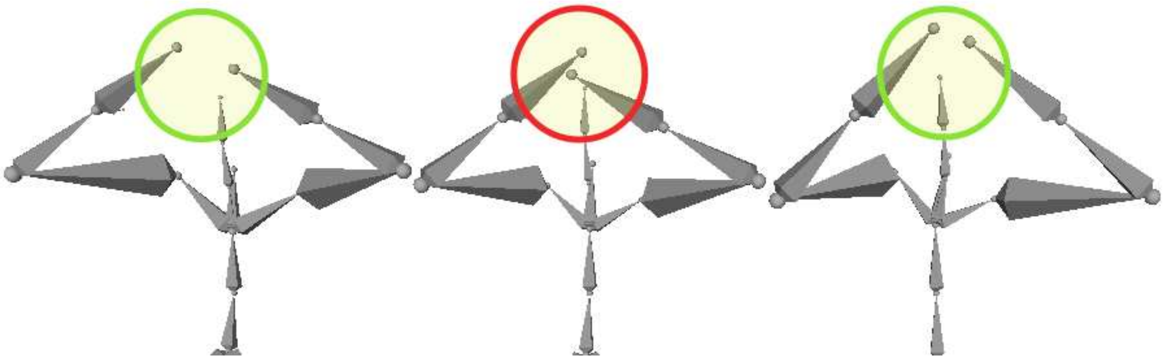


Fig. 2. A key frame for the macarena dance. From left to right: the posture in the original frame; the posture modified by simply transferring the original Euler angles to the different morphology; the posture obtained by our distance-based motion adaptation. The shoulders are 30% shorter in the target character. The distance between the two hands and their distance to the head are well preserved in our solution.

angles (because three of them are necessary to reconstruct the motion for each joint), one vector angle, alone, cannot be used to fully represent the motion of v. We couple therefore the ζ_v angle with a torsion angle.

Definition 2 *Given a skeletal structure (G,χ) and one realization x, the torsion angle ω_v for the joint v in this realization is the smallest angle formed by the plane defined by $x((p \circ p \circ p)(v))$, $x((p \circ p)(v))$ and $x(p(v))$, and the plane defined by $x((p \circ p)(v))$, $x(p(v))$ and $x(v)$.*

In the hypothesis the realization x preserves every bone length $||\chi(v)||$ (such as in the realization constructed in Equ. (1)), then, for every joint v having at least 3 ancestors, we can identify its position in space by using the information about its distance $||\chi(v)||$ to the parent, the vector angle ζ_v and the torsion angle ω_v [11]. Again, three real-valued variables come to play per joint. However, in our case, the distance $||\chi(v)||$ is supposed to be constant in the skeletal structure (G,χ), reducing in this way the degrees of freedom per joint to 2.

In the following, we will refer to our representation, which makes use of the two angles ζ_v and ω_v, as the "vector-torsion angle representation". Ours is not a completely original representation. The two angles ζ_v and ω_v are commonly used in the context of structural biology to represent molecular conformations [2]. Exclusive vector-torsion angle representations are bijective, in the sense that any representation with a different set of angle values corresponds to a different conformation, and vice versa. When comparing with this different biological application, we can notice that the underlying graph G has essentially the same features, even if its vertices represent *atoms* and not joints, while the morphology reflects the *chemical bonds* between pairs of atoms in the molecule. There are however two important differences in the two applications. First, in the "classical" backbone representation for special molecules named proteins, the vector angle ζ_v is also fixed (and not only the bond length $||\chi(v)||$), resulting in one unique degree of freedom per each quadruplet of consecutive atoms of the protein. In order to allow for wider movement possibilities, however, we cannot impose this constraint to our animated characters.

Secondly, and more importantly, molecular conformations do not need, in general, to be anchored to the environment. Several software tools for molecular visualization actually exploit this fact to allow the user to translate and rotate the molecular conformations as one pleases. This is not the case for our animated characters. As described in Section II, the fictive vertex $v_0 \in V$ is used for encode the displacement, as well as the rotation, of the character w.r.t. the origin of its environment.

Therefore, we propose to combine this vector-torsion angle representation of the character with the Euler-angle representation, and to use one or another representation depending on the vertex $v \in V$. If v is the only child of v_0, for example, there are two main reasons for choosing the triplet of Euler angles for its representation: first of all, v does not have enough ancestors for the vector-torsion representation; secondly, the triplet of Euler angles can provide the global orientation of the joint in the character environment.

Let $R \subset V$ be the subset of joints that admit less than three ancestors. We propose the use of the following function for the representation of character motions:

$$\rho' : (v,t) \in V \setminus \{v_0\} \times T \longrightarrow \begin{cases} \left(\theta^t_{p(v),v}, \phi^t_{p(v),v}, \eta^t_{p(v),v}\right) \in [0,2\pi)^3, & \text{if } v \in R, \\ (\zeta^t_v, \omega^t_v) \in [0,2\pi)^2, & \text{if } v \in V \setminus R. \end{cases}$$

Fig. 1 depicts an example of skeletal structure for a human character (the joint v_0 does not appear in the figure). With the different colors, we have marked the joints and bones involved in the definition of the vector and torsion angles. For example, the position of the joint named LeftFoot (see the part marked in green in the figure) can be reconstructed by using the constant length of the bone joining the LeftFoot with its parent joint, plus the vector and torsion angles obtained by taking into consideration the other two immediate ancestors. The total number of degrees of freedom is $3(|R|-1)+2(|V|-|R|)$, which is smaller and at most equal to $3(|V|-1)$ when no leaves in G admit at least 3 ancestors.

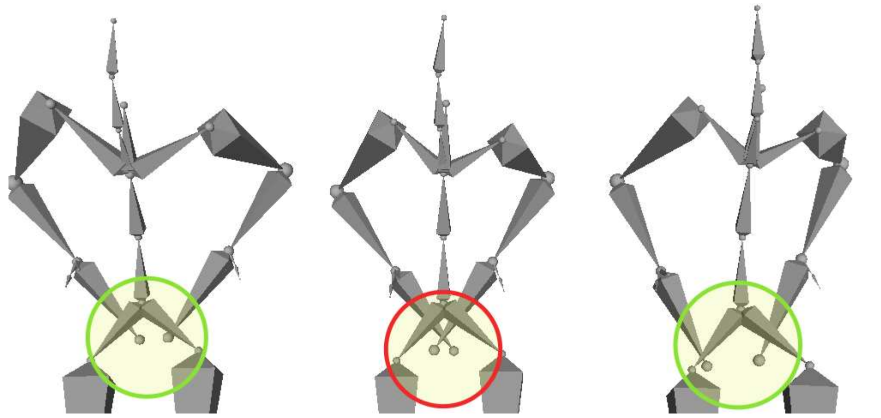


Fig. 3. Another key frame for the macarena dance. From left to right: the posture in the original frame; the posture modified by simply transferring the original Euler angles to the different morphology; the posture obtained by our distance-based motion adaptation. The shoulders are 30% shorter in the target character. The distance between the two hands does not exactly correspond to the original distance in our solution, but overall set of local distances around the character hands are well preserved, allowing the viewer to essentially perceive the same posture.

IV. The distance-based approach to motion adaptation revisited

In motion adaptation [6], the interest is in adapting an original motion, given for the character (G,χ), to a character having a different morphology $(G,\hat{\chi})$ (notice that the anatomy G is a constant). In this section, we will use the popular function ρ to represent the character motions (see Section II). In order to adapt the motion to the character $(G,\hat{\chi})$, a new function $\hat{\rho}$ needs to be defined, where the set of Euler angles, related to each character joints, are adapted to take into consideration the modifications in the character morphology, while trying to preserve as much as possible the original motion.

This work develops on the distance-based approach for motion adaptation previously presented in [3], [17], where dynamical distance geometry plays a main role [19], [23]. We exploit the set of inter-joint distances extracted from the original motion ρ of (G,χ) to control the proximity of the joints that are not connected by a bone in G. These distances allows us in fact to verify whether there are existing joints contacts in the original motion; and whether new (undesired) self-contacts can potentially be introduced while adapting the motion.

One of the main extensions w.r.t. the initial version of this distance-based approach consists in realizing the obtained motions directly in the space of Euler angles. In technical terms, instead of computing the coordinates, over time, for all the vertices of G in $\mathbb{R}^3$ (see Equ. (1)), it is now possible to directly define the function ρ to represent the adapted motions. Bond length fluctuations, observed in the experiments in [3], are in this way avoided by definition of the Euler-angle representation.

The first step in our distance-based approach consists in generating a dynDGP instance for which the solutions are the adapted motions [16], [17]. The distance information is extracted from the original motion ρ, as well as from the target morphology $\hat{\chi}$. From ρ, all inter-joint distances, at every frame $t \in T$, can be computed, including the original bone lengths $||\chi(v)||$. Even if the dynDGP is NP-hard in general [21], it is trivial to reconstruct, from this set of distances evolving over time, the original function ρ (modulo equivalent Euler-angle representations) [10]. Naturally, our interest is rather in constructing a new function, corresponding to the animation adapted to the morphology of the target character.

One initial manipulation that one can consider to perform, in distance space, consists in replacing all bone lengths with the new values $||\hat{\chi}(v)||$. However, this simple modification is likely to introduce large errors in the other involved distances, the ones that actually describe the motion. Therefore, while changing the bone lengths, we also modify accordingly other distances that can be computed from the original motion. We follow the procedure detailed in [17]. We initially compute all shortest paths $P_{uv} = \{w_1,\dots,w_k\}$ between pairs of distinct vertices, where $w_1 = u$, $w_k = v$ and, for every $i = 1,\dots,k-1$, we have $\{w_i,w_{i+1}\} \in E$ (notice that the term "shortest" makes reference to the number of edges that need to the crossed by the path to walk from the vertex u to the vertex v of the graph). We refer to the sum of the edge weights (bone lengths) over a path P_{uv} as the "weight" τ_{uv} of the shortest path P_{uv}, computed as:

$$\tau_{uv} = \sum_{i=1}^{(|P_{uv}|-1)} ||\chi(w_i) - \chi(w_{i+1})||.$$

The dynDGP instance is represented by a simple weighted undirected graph $H = (V \setminus \{v_0\} \times T, E_H, d)$, where d is composed by the two real-valued functions, the δ function, and the π function. The former function associates a distance to the subset of vertex pairs in H, as follows:

$$\delta : \{\{u,q\},\{v,t\}\} \in E_H \longrightarrow \begin{cases} \dfrac{\hat{\tau}_{uv}}{\tau_{uv}}||x_u^t - x_v^t||, & \text{if } t = q, \\ ||x_u^q - x_v^t||, & \text{if } u = v \text{ and } q = t-1, \end{cases}$$

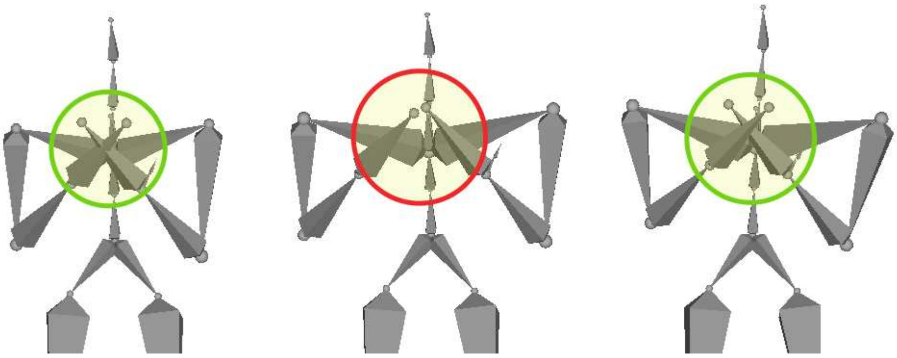


Fig. 4. A key frame for the character feeling cold and hence rubbing its hands. From left to right: the posture in the original frame; the posture modified by simply transferring the original Euler angles to the different morphology; the posture obtained by our distance-based motion adaptation. The shoulders are 30% longer in the target character. In the angle-transfer solution, the two hands can hardly touch one another.

where x_v^t represents the position of the joint v at time t indicated by the original function ρ. For all situations that are not considered in the definition of ρ (such as for example $q = t - 2$), we suppose that the corresponding edge is not included in E_H. We point out that the inter-frame distances were originally not taken into consideration in the distance-based approach: this is another novelty introduced in our revisited version.

The π function indicates the importance (the *priority*) of the distances given by the δ function. The first criterion to define the distance priorities is based on the fact that distances between joints that are closer in the skeletal structure (i.e. corresponding to shortest paths P_{uv} over fewer bones) can be approximated better than others. For example, bone lengths can be exactly transformed into the ones of the target morphology $\hat{\chi}$, while the distances between two leaves of G (such as a hand and a foot) rather represent rough approximations of the actual distances.

Our revisited approach improves this priority calculation by exploiting the information given by the *interaction distance* [15] between two joints u and v at a certain time frame $t \in T$. The interaction distance allows us to predict the distance that the two joints *will have* if their current relative movement (computed by comparing the current joint positions with the positions of the same joints at the previous frame) will not change in the subsequent frames. When two joints are moving one towards the other, their relative distances over time are in fact important for performing the adaptation, because they can guide the movement towards a joint contact that we want to preserve, or to avoid potential self-contacts in approaching joints. Therefore, we also assign a higher priority to the distances between joints u and v for which the corresponding interaction distance $I(u, v, t)$, at frame t, is smaller than a given positive threshold Δ.

Finally, we give the maximal importance (priority 1) to the newly introduced inter-frame distances. To sum up, our π function has the following form:

$$\pi : \{\{u, q\}, \{v, t\}\} \in E_H \longrightarrow \begin{cases} 1, & \text{if } q = t - 1, \\ 1, & \text{if } q = t \text{ and } (u = p(v) \text{ or } I(u, v, t) < \Delta), \\ (P_{\max} - |P_{uv}| + 2)/P_{\max}, & \text{otherwise,} \end{cases}$$

where $P_{\max}$ is the maximal length (in terms of number of crossing edges) for a path on the graph H.

In this work, we find solutions to the dynDGP instance represented by the created graph H by solving an optimization problem whose *stress* function measures the violation on the given distance constraints, where the violations on distances having higher priority give a higher contribution to the stress value (see [19] for a detailed description). We remark that, even if our current implementation is based on the Euler-angle representation ρ, it is necessary to compute the absolute positions x_v^t of every joint at every frame in order to calculate the value of the stress function. Since our dynDGP instance can be easily separated in several sub-instances representing the postures over time of our animated characters, we make the choice initially proposed in [17] to optimize the stress function frame by frame, by considering only the local distance information, and by exploiting the result obtained during the optimization process at the previous frame.

We run a non-monotone spectral projected gradient method for performing the adaptation of every frame of the motions by using the available distance information. Since, in the motions, the character postures slightly change from one frame to the subsequent one, the spectral method takes as a starting point the posture obtained at frame $t - 1$ to optimize the stress function for the current frame t. The only exception is given by the very first frame, where we select as a starting point the first frame of the original motion.

The next section presents some computational experiments performed with our revisited distance-based approach.

A. Distance-based motion adaptation in practice

We implemented our distance-based approach to human motion adaptation in the Java programming language. In this

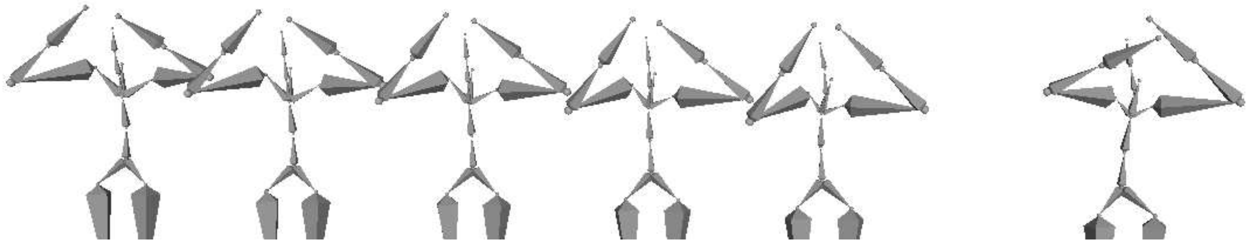


Fig. 5. On the right-most image, the result of retargeting the macarena dance key frame shown in Fig. 2 when the shoulders are 50% shorter. The sequence on the left-side of the figure shows the 5 steps to move from a 10% modification to the final 50% modification, with a change of 10% per step, necessary to obtain a good-quality solution when the changes in the morphology are more important.

section, we present some computational results where we have selected two motions from the Graphics Lab Motion Capture Database,[1] provided by the Carnegie Mellon University. Our Java code accepts a BVH file in input, containing the original motion, and outputs the retargeted motion in the same format. The pictures presented in this article were created with the free software Blender[2]. We will not report the numerical values of the stress function for the frames of the obtained motions because they do not always reflect the viewer perception on the correctness of postures or movements.

The first motion that we consider is the famous “macarena” dance (database entry code 135_35, see Fig. 2). At a certain point, the character is supposed to place its both hands on its head. The original frame is shown in the left-most image in Fig. 2. We can notice that, when the shoulders are reduced in length (-30%) and the original Euler angles are simply transferred to the new character (see central image), the two hands approach too much, and if one imagines where the head of the character is supposed to be, the viewer has the impression that the hands actually penetrate the character head. In the right-most image in Fig. 2, our solution shows a correct adaptation of the Euler angles to preserve the inter-joint distance between the hands.

In the same motion, the hands of the character are placed on its back a few frames later. The same change in the morphology implies another undesired effect in the animation that is obtained by simply transferring the Euler angles. Fig. 3 compares the original, the angle-transfer result and the result obtained with our approach. Again, one may well confuse the original posture with our result if they were not simultaneously visible to the viewer.

The next motion shows a character that feels like it is cold: to warm up the hands, it rubs them together (entry code 79_68, see Fig. 4). When the new character with shorter shoulders tries to warm up the hands, its two hands are actually too far for performing a proper rubbing movement in the angle-transfer posture (see central image in Fig. 4). The right-most image shows instead that this artifact is not present in our retargeted motion.

All adapted motions can be viewed on the YouTube video[3] of a talk given, a few months before the publication of this ar-

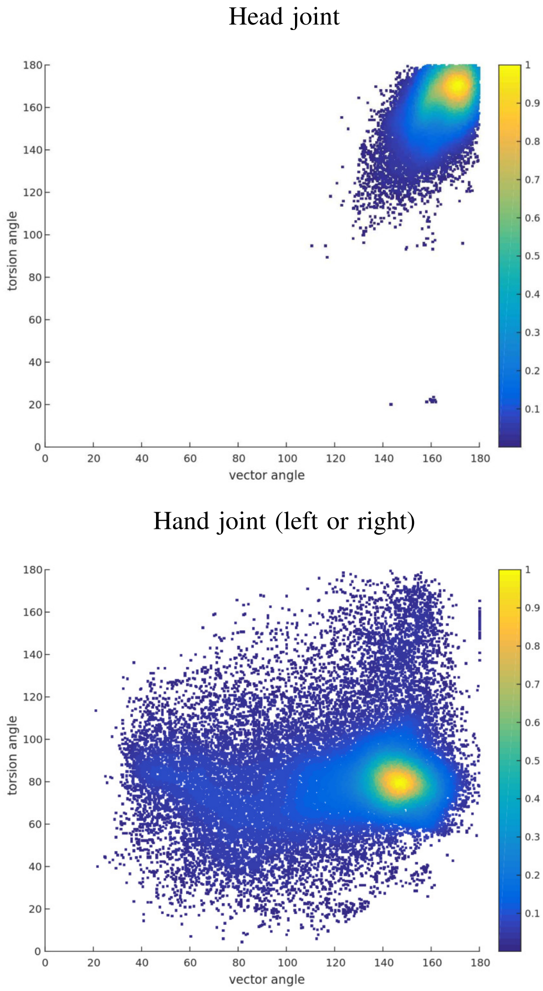


Fig. 6. Two scatterplots depicting the values of vector and torsion angles found in the representations of the Head and Hand joints. Points tending to the warmer colors correspond to pairs of angles that were found more frequently. The analysis reflects that the human hand is much more flexible than the head.

[1] https://mocap.cs.cmu.edu

[2] https://www.blender.org

[3] https://www.youtube.com/watch?v=UWf7z1cDWmc

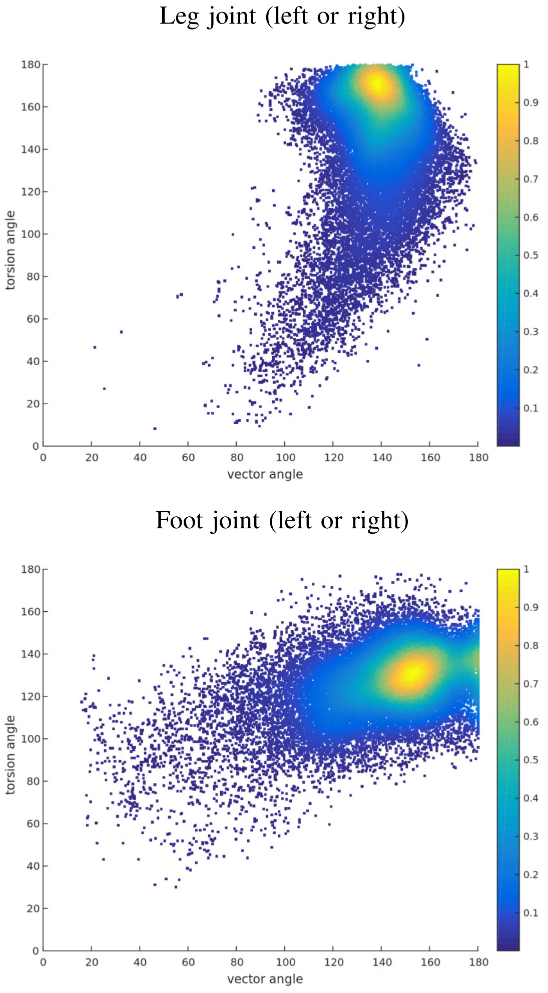
ticle, at a mini-symposium[4] focusing, among the other topics, on dynDGP applications. The reported animations (the same examples discussed here are presented) show that fluid motions can be obtained by our revisited distance-based approach.

Finally, one may wonder whether we can still obtain such good results when the modifications on the bone lengths are more important (more than 30% modification of the original length). In this case, as expected, the results get worse and worse, in general. In order to better deal with these more important changes, we have implemented an intermediate skeleton approach (see Fig. 5). The idea of using intermediate skeletons to improve the results of motion retargeting was initially proposed in [14] in an inverse kinematics approach; we have simply re-implemented it in our contest. Instead of attempting to retarget a motion with large morphology changes, the idea is to perform intermediate retargetings in a sequence, in order to *smoothly* approach to the desired morphology. Every skeleton in the sequence has an intermediate morphology between the original χ and the target $\hat{\chi}$. In the experiment depicted in Fig. 5, we can see that the intermediate skeleton approach can actually improve the quality of our retargetings when the changes on the morphology are more important.

V. An analysis on a database of human motions

In this section, we present a statistical analysis on the vector and torsion angles that we used in the new character representation presented in Section III, through the function ρ'. To perform the analysis, we consider a similar idea proposed in [20] for analyzing the two main torsion angles involved in the conformation of an amino acid in a protein backbone. In the studied molecules, in fact, the vector angles are considered as constants (this is a hypothesis that does not apply to animated characters, see discussion in Section III). Here, instead of studying subsets of atoms (the ones belong to an entire amino acid), we focus our attention on single joints, where we compare (we plot) the possible vector angles, related to the joint, against the corresponding torsion angles.

To perform our analysis, we consider the same database of human motions from which we selected the motions for our experiments in Section IV-A. This database consists of many motions with varying lengths (in terms of frames). Instead of preforming the analysis on the full motions of the database, we randomly selected 1% of the frames composing all the motions present in the database. We repeated this analysis several times to verify that the results were representative of the complete database. A useful observation from the analysis is that some joints have rather limited movement possibilities while others are much more flexible. Moreover, the analysis shows that simple constraints can potentially be introduced on the vector and torsion angles to constrain the characters to take only natural postures. This can potentially help our distance-based approach avoiding unnatural postures, especially when the changes in the morphology are more important.


Leg joint (left or right)

Foot joint (left or right)

Fig. 7. Two other scatterplots related to the Leg and Foot joints.

In Fig. 6, we compare the plot corresponding to the angles obtained while analyzing the Head joint to the plot obtained for the Hand joint. As one may have expected, the figure shows that the human hand has a quite large flexibility (almost the entire space is feasible), but we can identify a sub-region (in warmer colors in the figure) where the pairs of angles are more frequent. The joint representing the head of the character has instead much more limited movement possibilities. In the hypothesis the considered motion database actually covers all human suitable movements, we can therefore impose rather tight constraints to joints (such as the Head joint) in the ρ' representation, in order to ensure that the corresponding movements always look natural.

Fig. 7 shows two other plots, one obtained by analyzing the first joint representing the character legs (the parents are the hips), and another obtained by analyzing the character feet. Both leg and feet joints, on the left or right side of

[4] Mini-symposium on Sensor Network Localization and Dynamical Distance Geometry, scheduled as part of Thematic Program on Geometric Constraint Systems, Framework Rigidity, and Distance Geometry, Fields Institute, Toronto, Canada, May 18–27, 2021.

the character, define similar plots. Again, these plots present several combinations of angles (the white regions in the plot) that are never reached.

VI. Conclusions

We have revisited a distance-based approach to human motion adaptation, and presented a statistical analysis to establish the joint movements that allow the character to perform natural motions. To this purpose, we have discussed some alternative representations for the character motions. While we used the most popular representation (based on Euler angles) in our experiments on motion adaptation, we have found out that this may not be the most efficient one, and proposed an alternative representation, which we then used in our statistical analysis.

Future works will include a wider testing of this novel approach, with more human motions, and with several simultaneous changes (even asymmetric) performed on the morphology of the target character. Moreover, we will perform a deeper study on the several representations of the motions presented in this article, and we will compare each against the other. Currently, our results seem to suggest that the most efficient representation is the one introduced in Section III, and the result of our statistical analysis seem to validate this fact. However, this efficiency is currently supported, from a theoretical point of view, only by a simple study on its degrees of freedom, and therefore additional investigations in this direction need to be performed.

Acknowledgments

We are grateful to the anonymous reviewers for their comments on the initial version of this manuscript. This work is partially supported by the international project MULTI-BIOSTRUCT funded by the ANR French funding agency (ANR-19-CE45-0019).

References

[1] A. Alfakih, *Universal Rigidity of Bar Frameworks in General Position: a Euclidean Distance Matrix Approach*. In: [18], Springer, 3–22, 2013.

[2] H. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. Shindyalov, P. Bourne, *The Protein Data Bank*, Nucleic Acids Research **28**, 235–242, 2000.

[3] A. Bernardin, L. Hoyet, A. Mucherino, D.S. Gonçalves, F. Multon, *Normalized Euclidean Distance Matrices for Human Motion Retargeting*, ACM Conference Proceedings, Motion in Games 2017 (MIG17), Barcelona, Spain, November 2017.

[4] J. Diebel, *Representing Attitude: Euler angles, Unit Quaternions, and Rotation Vectors*, Matrix **58**(15–16), 1–35, 2006.

[5] R. Featherstone, *Rigid Body Dynamics Algorithms*, Springer, 279 pages, 2008.

[6] M. Gleicher, *Retargetting Motion to New Characters*. ACM Proceedings of the 25^{th} annual conference on Computer Graphics and Interactive Techniques, 33–42, 1998.

[7] S. Guo, R. Southern, J. Chang, D. Greer, J.J. Zhang, *Adaptive Motion Synthesis for Virtual Characters: a Survey*, The Visual Computer **31**(5), 497–512. 2015.

[8] E.S.L Ho, T. Komura, C-L. Tai, *Spatial Relationship Preserving Character Motion Adaptation*, Proceedings of the 37^{th} International Conference and Exhibition on Computer Graphics and Interactive Techniques, ACM Transactions on Graphics **29**(4), 8 pages, 2010.

[9] G. Laman, *On Graphs and Rigidity of Plane Skeletal Structures*, Journal of Engineering Mathematics **4**(4), 331–340, 1970.

[10] L. Liberti, C. Lavor, N. Maculan, A. Mucherino, *Euclidean Distance Geometry and Applications*, SIAM Review **56**(1), 3–69, 2014.

[11] T.E. Malliavin, A. Mucherino, M. Nilges, *Distance Geometry in Structural Biology: New Perspectives*. In: [18], Springer, 329–350, 2013.

[12] W. Maurel, D. Thalmann, *Human Shoulder Modeling Including Scapulo-Thoracic Constraint and Joint Sinus Cones*, Computers & Graphics **24**, 203–218, 2000.

[13] M. Meredith, S. Maddock, *Motion Capture File Formats Explained*, Technical Report 211, Department of Computer Science, University of Sheffield, 36 pages, 2001.

[14] J.-S. Monzani, P. Baerlocher, R. Boulic, D. Thalmann, *Using an Intermediate Skeleton and Inverse Kinematics for Motion Retargeting*, Computer Graphics Forum **19**(3), 11–19, 2000.

[15] A. Mucherino, *Introducing the Interaction Distance in the context of Distance Geometry for Human Motions*, Chebyshevskii sbornik **20**(2), 263–273, 2019.

[16] A. Mucherino, D.S. Gonçalves, *An Approach to Dynamical Distance Geometry*, Lecture Notes in Computer Science **10589**, F. Nielsen, F. Barbaresco (Eds.), Proceedings of Geometric Science of Information (GSI17), Paris, France, 821–829, 2017.

[17] A. Mucherino, D.S. Gonçalves, A. Bernardin, L. Hoyet, F. Multon, *A Distance-Based Approach for Human Posture Simulations*, IEEE Conference Proceedings, Federated Conference on Computer Science and Information Systems (FedCSIS17), Workshop on Computational Optimization (WCO17), Prague, Czech Republic, 441–444, 2017.

[18] A. Mucherino, C. Lavor, L. Liberti, N. Maculan (Eds.), *Distance Geometry: Theory, Methods and Applications*, 410 pages, Springer, 2013.

[19] A. Mucherino, J. Omer, L. Hoyet, P. Robuffo Giordano, F. Multon, *An Application-based Characterization of Dynamical Distance Geometry Problems*, Optimization Letters **14**(2), 493–507, 2020.

[20] G.N. Ramachandran, C. Ramakrishnan, V. Sasisekharan, *Stereochemistry of Polypeptide Chain Configurations*, Journal of Molecular Biology **7**, 95–104, 1963.

[21] J. Saxe, *Embeddability of Weighted Graphs in k-Space is Strongly NP-hard*, Proceedings of 17^{th} Allerton Conference in Communications, Control and Computing, 480–489, 1979.

[22] G.G. Slabaugh, *Computing Euler Angles from a Rotation Matrix*, Technical Report, City University London, 8 pages, 1999.

[23] P. Tabaghi, I. Dokmanić, M. Vetterli, *Kinetic Euclidean Distance Matrices*, IEEE Transactions on Signal Processing **68**, 452–465, 2020.

DOI: 10.15439/2021F60
ISSN 2300-5963 ACSIS, Vol. 25

Dynamic communication topologies for distributed heuristics in energy system optimization algorithms

Stefanie Holly
R&D Division Energy
OFFIS - Institute for Information Technology
Escherweg 2, 26121 Oldenburg, Germany
Email: stefanie.holly@offis.de

Astrid Nieße
R&D Division Energy
OFFIS - Institute for Information Technology
Escherweg 2, 26121 Oldenburg, Germany
Email: astrid.niesse@offis.de

***Abstract*—The communication topology is an essential aspect in designing distributed optimization heuristics. It can influence the exploration and exploitation of the search space and thus the optimization performance in terms of solution quality, convergence speed and collaboration costs – relevant aspects for applications operating critical infrastructure in energy systems. In this work, we present an approach for adapting the communication topology during runtime, based on the principles of simulated annealing. We compare the approach to common static topologies regarding the performance of an exemplary distributed optimization heuristic. Finally, we investigate the correlations between fitness landscape properties and defined performance metrics.**

I. Introduction

DISTRIBUTED heuristics are a promising field for current and future energy systems control and optimization tasks, and have been designed and evaluated in recent years on agent-based systems [1] [2] [3]. While conventional control systems – centralized or hierarchical in their control paradigm – perfectly fit to centralized generation and transmission systems, distributed renewable energy systems show properties that promote the application of distributed optimization systems: First, future energy systems can be regarded as complex systems of systems, sometimes framed as cyber-physical multi-energy systems, coupling communication systems, power, heat and gas systems. The resulting complexity of the solution space is the main motivation for heuristic distributed control and optimization [4]. Second, data availability as needed for centralized control typically is not given for end-user scenarios for privacy or regulatory reasons.

Distributed control and optimization systems often involve multiple energy units that decide locally and communicate with each other to solve global problems. For instance, software agents can represent flexible energy loads that cooperatively aggregate flexibility to provide load dispatch options for balancing markets or congestion management [5] [6] [7].

One major design aspect is the communication topology. This topology - usually modeled as a graph - determines which units exchange data directly. In energy system applications, the communication topology of multi-agent systems is often defined based on the topology of the underlying power grid (see e.g. [8] and [9]). This approach is limited to static topologies, and not reflecting algorithmic aspects. Considerable research has been conducted on distributed optimization in the power grid, in the area of control theory, where systematic mathematical approaches are used to design distributed controllers [10]. However, the type of problems that can be solved with such approaches is limited [11]. Distributed energy resources are very heterogeneous regarding forecast precision and flexibility potential. Agents have to consider more local constraints and be able to react flexibly to their environment. Thus, we consider algorithms that provide a framework for negotiation between different agents, but allow flexible local action. To the best of our knowledge, there has been little research on how the communication topology affects such distributed heuristics or how it can be optimally designed.

In [12] we showed that different communication topologies have an effect on the performance of the reflected algorithm class: Highly meshed topologies converged into good solutions reliably and quickly, but increased communication overhead and premature convergence. In contrast, results for sparsely meshed topologies were much less reliable. In the application domain of energy systems as critical infrastructures, this behavior is highly unwanted. We presume that dynamically adjusting the topology during runtime leads to a beneficial transition of exploration and exploitation of the search space for distributed heuristics.

In this contribution, we evaluate the effect of dynamic communication topology adaptation on a fully distributed optimization heuristic. To ensure scientific comprehensibility and reproducibility, standard optimization problems are taken for an extensive analysis of the approach. Furthermore, we analyse correlations between the performance of the distributed optimization algorithm and the fitness landscape characteristics of these benchmark functions with both static and dynamic overlay topologies using decision trees.

The rest of this contribution is structured as follows: In section II, an overview on the topic of communication topologies for distributed heuristics is presented, motivating the research gap. The dynamic topology adaptation scheme is presented in section III. The metrics used for the fitness landscape analysis are presented in section IV. In section V we set the scene for the experimental setup chosen to analyse the relevant correlations, followed by a discussion in section VI. We conclude our work with an outlook on future research directions.

II. Communication topologies for distributed heuristics

Distributed optimization heuristics are closely related to parallel cooperative metaheuristics [13]. In both types of heuristics, multiple distributed (meta)heuristics are interconnected and exchange information. Therefore, we consider the studies on the influence of exchange topologies in the area of parallel cooperative metaheuristics as relevant related work.

More precisely, we restrict our scope to asynchronous cooperative search strategies, i.e. several solvers run simultaneously (multi-search / distributed on algorithmic level) and cooperate with each other by asynchronously exchanging information. This type of parallel heuristics originates from the field of parallel computing. Therefore, the communication topology was mostly designed with respect to the hardware architecture (considering connections between processing units) leading to hypercube, ring or torus topologies [14]. For island model heuristics, i.e., heuristics where multiple instances of mostly population-based metaheuristics run in parallel exchanging individuals between their islands, a fully meshed topology is often chosen [15]. In addition, information is usually exchanged indirectly via a shared memory.

The effect of communication topologies on the performance of distributed optimization heuristics has been studied especially for the island-model, where the topology is often referred to as the migration topology. In most works, different topologies for a given parallel heuristic are studied on multiple benchmark problems and the topologies are ranked according to the achieved performance, which may involve different aspects [14], [16], [17], [18], [19]. Ruciński et al. investigated the effect of different migration topologies, including ring, cartwheel and hypercube topologies, on the performance of two different parallel global optimization algorithms cooperating via the island model [14]. They evaluated different topologies for both heuristics according to the performance obtained. Since the results varied widely, Ruciński et al. suggested that such studies be conducted in the future for other heuristics and with more problem instances.

Hijaze and Corne [16] analyzed how different topologies affect the performance of an asynchronous distributed evolutionary algorithm (EA). They evaluated the effect of the topologies on the performance of the algorithm using 30-dimensional target functions (Sphere, Rosenbrock, Schwefel, Rastrigin, Griewank, Ackley [20][21]). The success rate in finding the optimum was similar for all topologies, but better than for the standard single population EA (with equal total population size).

In their follow-up work in [17], they introduced an online adaptation of the migration scheme in which the migration probability was adjusted based on the progress of subpopulations on islands. With the adaptive scheme, optimal solutions were regularly found in less time and with a higher success rate, suggesting that a balance between exploration and exploitation can be achieved by dynamically adjusting the migration mechanisms.

In [18], Sanu and Jeyakumar conducted an empirical analysis on the performance of distributed differential evolution (DE) for varying migration topologies. They used various topologies (basic ring and ring variants, star, cartwheel, torus and mesh) and multiple benchmark functions (e.g. Sphere, Schwefel (1,2,3), Rosenbrock, Rastrigin) to investigate the impact of the topologies on the performance of an island model DE. They considered not only the convergence speed and solution quality based metrics, but also the computational effort, i.e., the number of function evaluations. They concluded that no single topology is suitable for all optimization problems and took a first step towards linking characteristics of the search spaces to the performance of the topologies by roughly categorizing the functions (modality and separability) and assigning the best performing topologies in each case.

The presented research can be summarized as follows: First, different communication topologies affect the performance of the various parallel metaheuristics. Second, the notion of performance is mainly limited to the achieved solution quality and convergence speed. Since the studies do not address spatially distributed systems, they usually do not examine the costs of collaboration, especially the resulting message traffic. Third, the design of communication topologies leads to different balances between exploration and exploitation of the search space. In some cases it is investigated how this balance can be improved by adjusting parameters like migration frequency or migration rate. However, the adaptation of the topology has not been treated as a distinct research topic for optimization problems with characteristics as be found in energy system applications. To our knowledge, a systematic approach regarding the above mentioned aspects including an in-depth fitness landscape analysis has not yet been conducted in this field.

III. Dynamic Topology Adaptation

As described in the previous section, the communication topologies of distributed optimization heuristics affect the degree of exploration and exploitation of the search space. Strongly meshed topologies lead to a high amount of information exchange between units (diversification). This leads to a fast convergence but bears the risk of a premature convergence into local optima. In contrast, with sparsely meshed topologies, the individual heuristics can evolve more independently. This leads to an intensified search (exploitation) in some areas of the search space. But in the worst case, these areas can be far away from the global optimum.

Many metaheuristics adjust parameters at runtime to allow a transition from exploration to exploitation. An example of this is the adjustment of the temperature parameter T for simulated annealing (SA) [22].

Since we want to achieve the same effect with dynamic topology adjustment, our approach is based on the principles the cooling process in SA. Just as SA starts with a high temperature, we start with a high number of connections in the communication topology. With cooling down, we reduce the number of connections. To model the cooling process, SA

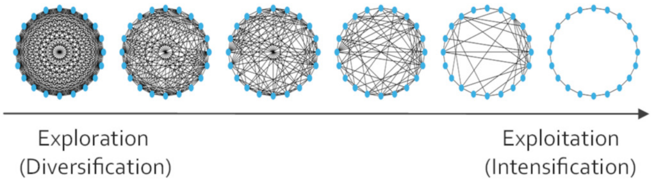


Fig. 1: Dynamic topology adaptation

uses a so-called cooling schedule. We therefore determine a so called "removal schedule". Table I shows the details of the specified analogy. The approach presented here starts with a fully meshed topology, transitions to small world intermediate stages by removing edges, and ends with a ring to exploit the most promising regions in the solution space. Fig. 1 illustrates this process.

In order to model this transition, more specifications are necessary. Let $G = (V, E)$ denote the bidirectional graph that represents the communication topology. V is the set of nodes, where each node is assigned to an agent and thus to one part of the distributed solver. E is the set of edges. An edge between two nodes indicates direct communication between the two agents assigned to the nodes. Agents can pass on information from their neighbors to other neighbors, which means that there is also indirect communication between unconnected nodes. The communication topology thus regulates the information dissemination in the distributed system.

A cooling schedule for SA is determined by the initial temperature T_0, the equilibrium state, i.e. the criterion that controls when the transition to the next temperature level occurs, and the cooling itself. Since the communication topology will start with a fully meshed bidirectional graph, the number of edges is defined as $|E_0| = \frac{n \cdot (n-1)}{2}$. Geometric functions are particularly popular to model cooling, whereas logarithmic functions are considered too slow for practical application, although they theoretically converge to a global optimum [13].

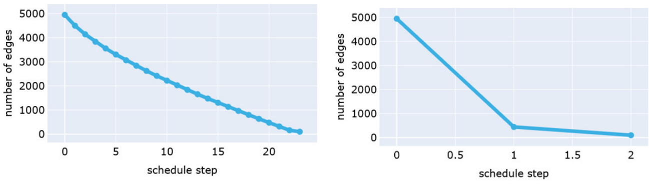
Considering that the initial number of edges is much smaller than usual starting temperatures, a slower reduction seems appropriate. A combination of linear, geometric and logarithmic reduction functions was chosen. Equation 1 displays the function that determines the number of edges δ at each schedule step.

$$\delta_{i+1} = |E_i| - \frac{|E_0|}{log(i)} \cdot \alpha, \quad with \;\; \alpha \in \left]0, 1\right] \tag{1}$$

The index i represents the index of the step in the reduction schedule. δ_i determines the number of edges that should remain in the reduction schedule step i. Thus the new communication topology graph is constructed such that

$$G_{i+1} = (V, E_{i+1}), \quad with \quad |E_{i+1}| = \delta_{i+1} \tag{2}$$

The parameter α controls how many steps the removal schedule includes. If it is close to 0, only a few edges are removed in each step, leading to a slow decrease of connectivity. If it equals 1, the number of edges is reduced in large steps, which leads to a rapid decrease in connectivity. The edges that are removed are selected randomly, ensuring that the final ring topology remains. The figures 2a and 2b show how the parameter α influences the granularity of the edge reduction schedule.



(a) $\alpha = 0.1$

(b) $\alpha = 1$

Fig. 2: Reduction schedule with 100 agents and different α

Finally, the transition criterion from one topology to the next must be specified. It defines when a transition from one schedule step to the next occurs. Common conditions for a temperature adaptation in SA are counting of iterations, acceptances, rejections or a combination of these [13]. In this case, a simple approach is used, where the number of local searches (equivalent to the number of iterations) since the last transition is counted. As soon as this number becomes larger than n, the topology is adjusted. Since the starting topology is a complete graph all agents start at the same time and each unit performs its local search at least once before the first adjustment. In later stages, this procedure can lead to some agents optimizing locally multiple times between topology adjustments and others not optimizing at all. This rapid transition was chosen since it showed the best results in preliminary experiments.

To evaluate this dynamic approach, we perform a systematic comparison of the performance of the approach in different parameterizations and several static topologies. For this purpose, we use a set of well-known benchmark functions. However, instead of considering them separately, we perform fitness landscape analysis to match the individual difficulties of the problems with the performance of the topologies and find possible correlations. Consequently, we first explain the metrics used in the following section before moving on to the experimental setup and the evaluation of the results.

IV. Fitness Landscape Analysis

As the no free lunch theorem states, no optimization heuristic can be superior to all others without regard to the problem [23]. Different communication topologies significantly influence the information propagation in a heuristic and thus the resulting optimization process. Consequently, for different problems, different topologies presumably lead to more advantageous behavior. We use various fitness landscape metrics to classify the objective functions to examine the relationships between problem characteristics and the performance of different topologies.

A fitness landscape, as presented e.g. in [24], is defined by the search space X, containing all possible solutions of the problem, connected according to a defined distance measure,

TABLE I: Comparison of the modeling of SA cooling and the connection reduction of the communication topology

	cooling schedule	removal schedule
definition of	temperature for each step of the SA algorithm	number of edges in the communication topology for each step
initialization parameter	T_0: initial temperature	$\lvert E_0 \rvert = \delta_0$: initial number of edges
equilibrium state	number of iterations at a temperature	number of local optimizations at a topology configuration
adaptation	cooling: decrease of the temperature	decrease of edges in the communication topology

and the fitness function $f : X \rightarrow R$. A fitness landscape for a continuous problem, often uses euclidean distance measures and thus can be described as a landscape with the search space as the bottom floor and the landscape surface being elevated according to the values of the fitness function f [13]. Analogous to a geographical landscape, fitness landscapes can have peaks, valleys, plains, canyons, cliffs, plateaus, basins, etc. Investigating these landscape characteristics provides clues as to how difficult it is to find an optimum, i.e., the highest mountain peak (maximization) or the lowest valley (minimization). Fig. 3 shows 3-D plots of benchmark functions demonstrating some manifestations of such landscape features.

Various metrics have been proposed in literature. We limit the scope to metrics that can be used for continuous search spaces and that can be normalized, since we want to be able to compare different functions. In [24], Sun et al. distinguished some basic features of fitness landscapes and argued that for proper characterization, these features must be covered when selecting a set of metrics. These features include:

- Dimensionality
- Separability
- Ruggedness, Smoothness and Neutrality
- Modality
- Deception and Evolvability

The dimensionality of the problems is a selectable parameter in the experimental setup and therefore known. Furthermore, separability is a well-known property of the benchmark functions. For the other characteristics mentioned, suitable metrics must be selected. We discuss our choice in the following.

A. Ruggedness, smoothness and neutrality

The characteristics of ruggedness and smoothness concern the quantity and distribution of the local optima in the search space. The fitness differences in a neighborhood can be large (rugged), small (smooth), or barely present (neutral). Each of these surface shapes presents different challenges for optimization algorithms. In [25] Malan and Engelbrecht adapted the entropy based measure for ruggedness that was first proposed by Vassilev et al. [26] for continuous fitness landscapes. The information theoretic technique is based on a random walk through the search space. The random walk is represented as a string with respect to the information stability measure ϵ. If the magnitude of the difference between two fitness values is less than ϵ, they are considered to be equivalent. The string representation is obtained as follows:

$$S_i(\epsilon) = \begin{cases} -1, & \text{if } f_i - f_{i-1} < -\epsilon \\ 0, & \text{if } |f_i - f_{i-1}| \leq \epsilon \\ 1, & \text{if } f_i - f_{i-1} > \epsilon \end{cases} \tag{3}$$

The entropy value is calculated for the resulting string $S_i(\epsilon)$ according to Equation 4:

$$H(\epsilon) = -\sum_{p \neq q} P_{[pq]} log_6 P_{[pq]} \tag{4}$$

where $P_{[pq]}$ is the frequency of occurrence of the block $[pq]$ in $S_i(\epsilon)$ with $p, q \in \{-1, 0, 1\}$. This entropy is calculated with various ϵ between 0 and ϵ_{max}, which is the value at which the resulting string consists only of zeros. The maximum of all attained entropy measures is taken as the final result [25]. This entropy measure reflects the information content of the random walk. This is naturally high for a rugged landscape, whereas a smoother landscape has a smaller entropy value.

Depending on the stepsize of the random walk, ruggedness can be viewed on different scales. We follow Malan's suggestion and compute the metric once based on random walks with a maximum step size of 1% of the search space and once with 10%. The resulting metrics FEM_{micro} and respectively FEM_{macro} (First Entropic measure as defined by [27]) are values in $[0, 1]$ and reflect the relationship between ruggedness and neutrality on micro an macro scale.

Similarly to the FEM a second entropy measure can be calculated that estimates the smoothness of the function rather than the ruggedness. Based on the string $S_i(\epsilon)$ the entropy of smooth blocks, i.e. two consecutive characters with the same sign, is calculated as follows:

$$h(\epsilon) = -\sum_{p = q} P_{[pq]} log_3 P_{[pq]} \tag{5}$$

We apply the same approach as for FEM by calculating $h(\epsilon)$ with different values for ϵ and keeping the maximum of all entropy values as SEM (Second Entropy Measure). The SEM_{micro} and SEM_{macro} are again values in $[0, 1]$, but refer to the interaction of smoothness and neutrality of the landscape [27].

B. Modality

The *modality* of a function corresponds to the number of local optima. Modality and ruggedness are closely related, since a rugged landscape may also include many local optima. In [26], Vassilev et al. also proposed a metric to quantify the modality of the random walk encoded by Equation 3. A new string $S'(\epsilon)$ is constructed by removing all zeros from $S(\epsilon)$ and reducing sequences of equal characters to one character. Thus, $S'(\epsilon)$ contains only information that is essential with respect to modality. The resulting modality measure is called partial information content (PIC) and is given by

$$PIC(\epsilon) = \frac{\mu}{n} \tag{6}$$

where n is the length of $S(\epsilon)$ and μ the length of $S'(\epsilon)$. If the random walk encountered a landscape with high modality, the length of $S'(\epsilon)$ is almost the same as that of $S(\epsilon)$, resulting in $PIC(\epsilon)$ being close to or equal to one. If the path is flat or only leading in one direction $PIC(\epsilon)$ tends to or is equal to zero. For PIC, we use the same procedure as for FEM and SEM and perform random walks with varying step sizes to look at modality at different scales. Accordingly, we use PIC_{micro} and PIC_{macro} as metrics.

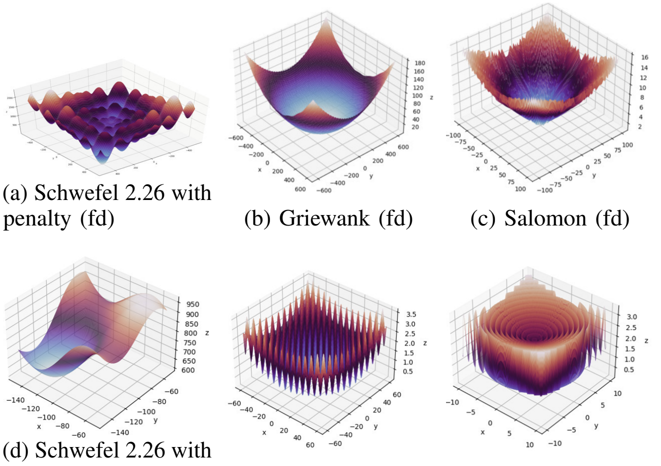
C. Deception and evolvability

A *deceptive* landscape provides information that can guide an optimization algorithm away from the global optimum, towards local optima. Which properties of a function are deceptive depends on the algorithm - in the case of distributed algorithms, perhaps also on the communication topology. One possibly deceiving characteristic is the presence of funnels. A funnel is a cluster of local optima that forms a global basin shape [28]. The dispersion metric of Lunacek et al. [29] provides insight into the global topology of fitness functions and thus indirectly allows estimation of the presence of funnels. In [28], Malan and Engelbrecht proposed a normalized version to allow comparison of functions with different domain sizes. To compute the dispersion metric, a random sample $\mathscr{S}$ of length n is drawn that is uniformly distributed over the search space. From this sample $\mathscr{S}$, a subset $\mathscr{S}^*$ is determined that contains the best points by fitness values. To make functions with different domain sizes comparable, the position vectors of $\mathscr{S}^*$ are normalized in such a way that the search space is scaled to [0,1]. In addition, a comparison sample $\mathscr{C}$ also of size is sampled uniformly across the search space. Let $disp(\mathscr{S})$ be the average pairwise distance between normalized positions in the sample $\mathscr{S}$. Then the dispersion metric DM is defined as follows:

$$DM = disp(\mathscr{S}^*) - disp(\mathscr{C}) \tag{7}$$

Thus, the metric quantifies how far points with high fitness values are away from each other compared to a large uniform random sample. It yields values in the range of [-1,1]. A low value ($DM < 0$) indicates a single funnel landscape with an underlying unimodal structure. A high value ($DM > 0$) indicates a multi-funnel landscape and underlying multimodal structure. Fig. 3 shows 3-d plots of three benchmark functions once in their full domain and once in a one-percent section of their domain, respectively. The penalized Schwefel 2.26 function is very rugged on the macro scale (Fig. 3a), but much less so on the micro scale (Fig. 3d). The function has a high value in the dispersion metric, which is also consistent with its global multi funnel shape. In contrast, both Griewank and Salomon have a small value in the dispersion metric, corresponding to their global single funnel shapes. While Salomon is highly rugged on macro and micro scale with a slight increase on micro scale, Griewank is much more rugged on micro scale than on the macro level. This effect actually decreases in higher dimensions, making the function "easier" to solve [30].

Fig. 3: Selection of benchmark functions as 3-D plot on the full domain (fd) or on a 1 % section of the domain (1%d)



(a) Schwefel 2.26 with penalty (fd)
(b) Griewank (fd)
(c) Salomon (fd)
(d) Schwefel 2.26 with penalty (1%d)
(e) Griewank (1%d)
(f) Salomon (1%d)

Table II shows an overview of the applied fitness landscape metrics, including a brief summary. The results obtained for the metrics for the benchmark functions are summarized in a dedicated repository[1].

dimension	dimension, here also equal to number of agents
separability	boolean that indicates if variables of a function are independent
FEM_{macro} FEM_{micro}	first entropy based measure of *ruggedness* on macro and micro scale
SEM_{macro} SEM_{micro}	second entropy based measure of *smoothness* on macro and micro scale
PIC_{micro} PIC_{macro}	measure of partial information content concerning *modality* on macro and micro scale
DM	dispersion metric that quantifies distances between good solutions and thus indicates the presence of funnels

TABLE II: Overview of applied metrics

V. Methodology

The goal of the experimental study is to investigate if a dynamic topology adaptation approach outperforms static topologies. We use multiple benchmark functions, all scalable and multi-modal, and the distributed optimization heuristic presented in subsection V-A.

Furthermore, we examine correlations between the properties of the benchmark functions and the performance of different

[1] https://github.com/sholly-offis/Deta

topologies in the defined performance dimensions. In doing so, we also consider different parameterizations of the dynamic approach. Thereby, we hope to find clues that will help in the further development of the dynamic approach and ultimately lead to the parameterization of dynamic topology adaptation in such a way that it provides a tailored solution to a problem. In the following we first give a short introduction to the chosen example heuristic and then elaborate on the experimental setup.

A. *Distributed Optimization Algorithm*

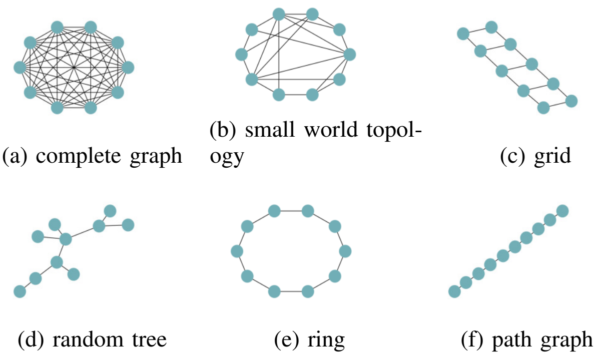
COHDA is a combinatorial optimization heuristic for distributed agents, and was developed for the self-organized scheduling of distributed energy resources in virtual power plants [31]. The heuristic can be classified as a system realizing a gossiping protocol based on strictly defined communication and knowledge integration rules. In [32], Bremer et al. adapted COHDA to find the global minimum of a real valued objective function. An agent a_i is responsible for only one value x_i from a continuous search space. It performs its local optimization to minimize the global objective function, by adapting its own choice of x_i while considering the choices of other agents $x_j, j \neq i$ as temporarily fixed. Agents send update messages to their neighbors - as defined by the communication topology - to pass on new information from their neighbors or to inform about their own changes in the value selection. Depending on well-defined convergence conditions, COHDA has been proven to always converge at least to a local optimum [31]. Parts of these conditions are related to the topology, and thus have to be reflected here: The chosen topology has to be connected, irreflexive, and symmetric. As a consequence, the topology adaptation developed in this work has to guarantee these characteristics in all intermediate stages to not sacrifice convergence.

B. *Experimental Setup*

The experiments are preformed with agent systems in two different sizes, namely 50 and 100. In the applied setup, the system size is equal to the dimension of the objective functions, as each agent is responsible for choosing one solution variable. A set of 13 different benchmark functions is used as underlying synthetical problem instances. They include Ackley, a scalable version of Eggholder, Griewank, Happy Cat [33], Rana, Rastrigin [21], Rosenbrock, Salomon, Schaffer F6, Qing, Schwefel 2.26 and a penalized Version of Schwefel 2.2.6 which was introduced in [12]. Unless otherwise stated, the definitions are taken from [20]. Definitions, domains and global minima of the benchmark functions are listed in a dedicated repository[2].

The search process in the solution space defined by each benchmark function is performed using COHDA and different topologies. This involves complete graphs, ring-, tree-, small-world-, path-, and grid- topologies. Fig. 4 shows these topology types for 10 agents.

Fig. 4: Overview of reflected static communication topologies



(a) complete graph

(b) small world topology

(c) grid

(d) random tree

(e) ring

(f) path graph

For each system size (50 and 100) each topology type is created, including a randomized setting with 5 different seeds and optimization runs and 10 different starting seeds. As a result of this setting, 50 different setups are examined for each combination of the number of agents and topology type. The same is done for the dynamic approach with different values for $\alpha \in [0.1, 0.3, 0.7, 1]$. [3] A total of 1000 optimization runs were performed per benchmark function (600 with static topologies and 400 with the dynamic approach). Therefore, the following evaluations are based on a total of 13,000 optimization runs.

To compare the performance, four performance dimensions are evaluated:

- solution quality: error measure
- speed of convergence: time required to converge
- communication traffic: number of messages exchanged between agents
- computational effort: number of local searches preformed by all agents

Each of the performance dimensions is normalized per system size and benchmark function. We differentiate the four performance dimensions when determining the best topology for a benchmark function. To be the best topology in a performance dimension, a topology must achieve the lowest possible values for this measure. Therefore, we consider the mean in each case. For some topologies however, the spread of values is extremely large, so that despite a good mean value, the risk of an unfavorable outlier is much higher than for other topologies. Again, in the application domain of critical infrastructure optimization, this is unfavorable. The sum of mean and standard deviation is used as an additional measure (using the notion *mps* – mean plus standard deviation), to reflect this aspect.

To relate the performance of the topologies to the characteristics of the benchmark problems, fitness landscape metrics described in section IV are computed for each benchmark function using random walks of length 1000 and the mean of 30 runs is taken as measure as proposed by [28]. Finally, we train decision trees using the CART algorithm [34] implemented by scikit-learn [35]. We use them to determine which metrics can be employed to distinguish benchmark functions and assign them to the best-performing topologies.

[2] https://github.com/sholly-offis/Deta

[3] $\alpha = 0.5$ was discarded in this presentation for reasons of brevity as it was not superior to the other values.

VI. Results and Discussion

In the following, we first examine the differences in the solution quality, using decision trees generated on the basis of the fitness landscape analysis. Additionally, we want to identify correlations between the properties of the fitness landscapes and the performance of different topologies and parameterizations of the dynamic adaptation approach.

In the second part of the evaluation, we examine the other performance dimensions. Since these cannot be reasonably analyzed on their own, individual functions are analyzed as representatives in order to investigate the overall performance of the different topologies and the trade-offs between the performance dimensions.

A. Decision tree-based performance analysis

To determine which topology is best for a given function and dimension for a performance indicator, minimum mean and mps are evaluated in each case. For simplicity, additional collective topology categories were introduced when several topologies were equally good. All topologies whose mean and mps do not deviate more than 1% from the minimum values are assigned to a best list. If the best list contains more than three entries, single topologies are replaced by collective categories. The label *highly meshed* is assigned if only highly meshed topologies are in the given class (complete, small world, grid, dynamic). *Weakly meshed* accordingly summarizes ring, tree, and path graph topologies. Other combinations are labeled as *various*. Note that if $dynamic$ is displayed without a value for α, multiple values for α have performed equally well. In order to classify what a decision boundary for a particular landscape metric means, the distribution of the metric in the set of benchmark functions must be considered. For this purpose, Table III shows the respective minima, maxima, mean values and the limits for the upper and lower quartiles.

	min	Q_1	median/Q_2	Q_3	max
DM	-0.39	-0.28	-0.2	-0.01	0.05
FEM_{macro}	0.69	0.72	0.87	0.88	0.89
FEM_{micro}	0.2	0.4	0.66	0.76	0.9
SEM_{macro}	0.51	0.53	0.57	0.68	0.74
SEM_{micro}	0.52	0.65	0.73	0.78	0.8
PIC_{macro}	0.35	0.38	0.61	0.66	0.7
PIC_{micro}	0.07	0.14	0.32	0.46	0.74

TABLE III: Distribution of fitness landscape metrics

First, we focus on the achieved solution quality in terms of the normalized error. Fig. 5 shows the resulting decision tree as sankey diagram for the mean error and Fig. 6 the tree illustrating the error mps. The thickness of the branches indicates the number of benchmark functions that belong to a path. On the right, the best topologies or topology categories are shown with the functions they contain. To facilitate the discussion of the results, individual end nodes or clusters of end nodes were numbered. These are discussed below.

1) functions with below average micro ruggedness (FEM_{micro}) and small dispersion (DM), i.e. good points in the search space are close to each other; These function are easy to solve, thus many topologies provide good results
2) Sargan and Happy Cat function; The Sargan function has the highest values for macro smoothness (SEM_{macro}). The Happy Cat function is also very smooth. For both functions complete graph and the dynamic approach perform similarly, while all other topologies rank far behind.
3) functions with below average micro ruggedness but higher dispersion; slightly more difficult than (1), thus weakly meshed topologies are outperformed.
4) functions with above average micro ruggedness (FEM_{micro}), above average macro smoothness (SEM_{macro}) and low to medium modality on macro scale (PIC_{macro}) are solved best by $dynamic_{\alpha=1}$ topology. Solomon (50) has a very high micro ruggedness. If only the mean error is considered, the ring topology performs best. However, if the reliability is also considered (see Fig. 6), the $dynamic_{\alpha=1}$ topology performs better. The ring topology is obviously very well suited for the intensive exploitation of landscape areas with high micro ruggedness and lower macro smoothness, but an initial exploration push, as in the dynamic approach, helps to guide the search more reliably into good regions.
5) functions with above average micro ruggedness and macro smoothness in the lower quartile; dynamic topology adaptation with different values for α achieves the best results there;
6) functions with above average micro ruggedness but very low smoothness on micro scale but not on macro scale; Considering only the mean error, the grid and tree topologies perform best, but their advantage is very small, so for the mps value for the Ackley function, many topologies perform equally well and Eggholder (100) is best and most reliably optimized with the dynamic topology with $\alpha = 0.7$.

Note that functions for which $dynamic_{\alpha=1}$ topology performs best, have the highest values for the dispersion metric. A DM larger than -0.037 seems to be a good indicator for a class of functions with multi-funnel shapes for which the dynamic approach with large values for α achieves the best results. Fitness landscapes with high dispersion, high modality, high ruggedness and low smoothness less strongly meshed topologies are advantageous, which often makes the dynamic approach with large values for α favorable. For landscapes with lower modality, less dispersion, a less rugged surface, and more smooth sections, more strongly meshed topologies tend to have an advantage, and the dynamic approach with small values for α also often performs well. More analysis on this will be needed in future work.

B. Examination of further performance dimensions

For the analysis of the other performance dimensions, an isolated consideration of the decision trees is not suitable, since there are usually correlations between good and poor

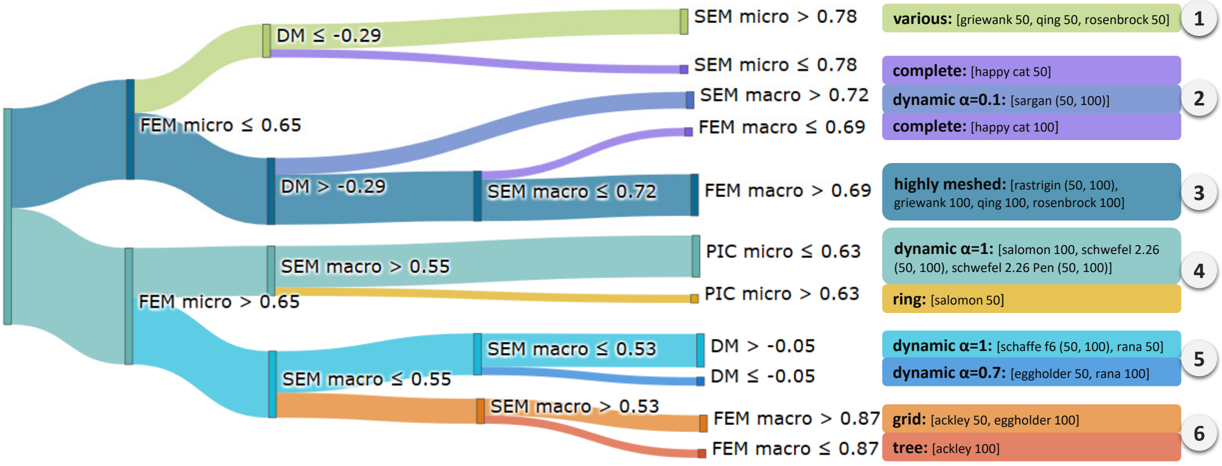

DM ≤ -0.29
SEM micro > 0.78
SEM micro ≤ 0.78
FEM micro ≤ 0.65
SEM macro > 0.72
FEM macro ≤ 0.69
DM > -0.29
SEM macro ≤ 0.72
FEM macro > 0.69
SEM macro > 0.55
PIC micro ≤ 0.63
FEM micro > 0.65
PIC micro > 0.63
SEM macro ≤ 0.53
DM > -0.05
SEM macro ≤ 0.55
DM ≤ -0.05
SEM macro > 0.53
FEM macro > 0.87
FEM macro ≤ 0.87
various: [griewank 50, qing 50, rosenbrock 50]
1
complete: [happy cat 50]
dynamic α=0.1: [sargan (50, 100)]
complete: [happy cat 100]
2
highly meshed: [rastrigin (50, 100), griewank 100, qing 100, rosenbrock 100]
3
dynamic α=1: [salomon 100, schwefel 2.26 (50, 100), schwefel 2.26 Pen (50, 100)]
ring: [salomon 50]
4
dynamic α=1: [schaffe f6 (50, 100), rana 50]
dynamic α=0.7: [eggholder 50, rana 100]
5
grid: [ackley 50, eggholder 100]
tree: [ackley 100]
6

Fig. 5: Decision tree for mean error

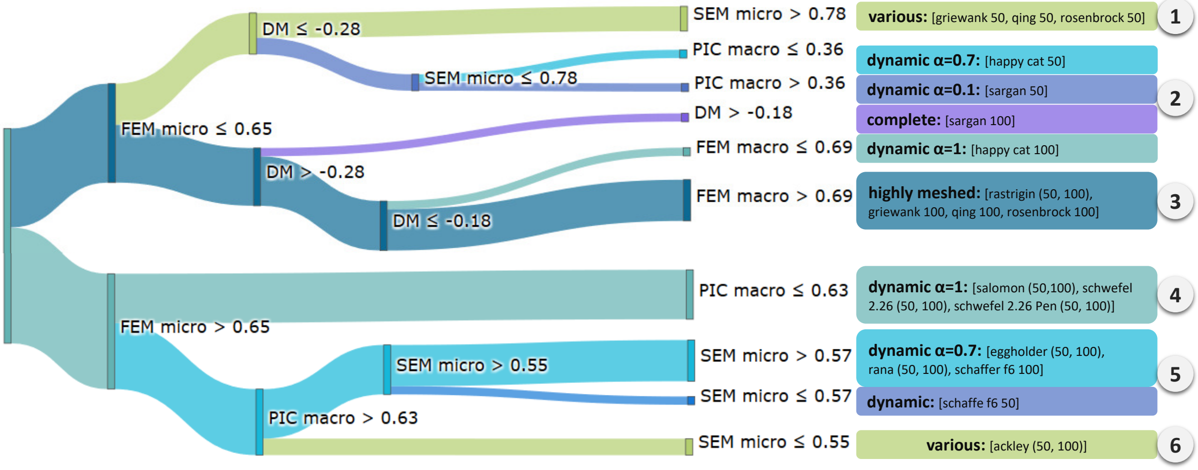

DM ≤ -0.28
SEM micro > 0.78
PIC macro ≤ 0.36
SEM micro ≤ 0.78
PIC macro > 0.36
DM > -0.18
FEM micro ≤ 0.65
FEM macro ≤ 0.69
DM > -0.28
DM ≤ -0.18
FEM macro > 0.69
PIC macro ≤ 0.63
FEM micro > 0.65
SEM micro > 0.55
SEM micro > 0.57
SEM micro ≤ 0.57
PIC macro > 0.63
SEM micro ≤ 0.55
various: [griewank 50, qing 50, rosenbrock 50]
1
dynamic α=0.7: [happy cat 50]
dynamic α=0.1: [sargan 50]
complete: [sargan 100]
dynamic α=1: [happy cat 100]
2
highly meshed: [rastrigin (50, 100), griewank 100, qing 100, rosenbrock 100]
3
dynamic α=1: [salomon (50,100), schwefel 2.26 (50, 100), schwefel 2.26 Pen (50, 100)]
4
dynamic α=0.7: [eggholder (50, 100), rana (50, 100), schaffer f6 100]
dynamic: [schaffe f6 50]
5
various: [ackley (50, 100)]
6

Fig. 6: Decision tree for mps error

performance in several dimensions. For example, the heuristic may have terminated after a short time, but converged to a local optimum. Or only very few messages are sent, but only small parts of the search space are explored, which makes the solution quality a matter of luck. Therefore, we examine the results for the penalized Schwefel 2.26 function, Rana, and Griewank in more detail below.

Fig. 7 shows the results obtained for Schwefel 2.26 with penalty for a system size of 100 agents. Each of the four performance dimensions considered is presented in a separate row. For each static topology and the dynamic approach with $\alpha = 1$ a violin plot is displayed, which shows the distribution with an internal box plot surrounded by a density plot. The top row shows the distributions of the normalized errors. For this function, the dynamic approach performs with the highest solution quality in terms of mean value and outliers. Compared to the strongly meshed topologies (complete, small world and grid graph), the computational effort depicted in row two of Fig. 7 is slightly increased. This seems reasonable, since the topology converges to a ring topology after a short time and therefore performs more computations (with the design objective of these computations being in a promising region of the search space). A similar picture is obtained for the emerging communication traffic, row three in Fig. 7. However, the complete graph generates much more communication effort. In terms of convergence time, the dynamic approach shows slower convergence compared to the topologies that are consistently strongly meshed.

For Rana, the overall impression is similar to Schwefel 2.26 with penalty. The dynamic approach usually performs best and thus needs more time, computational effort and message exchange than other strongly meshed topologies. If the main concern is convergence speed or low communication traffic, the small-word or grid topology are preferable as they provide a good trade-off. For functions with lower modality, dispersion, and ruggedness, such as Rastrigin or Griewank, where many topologies succeed in finding the global optimum, the complete and dynamic graphs usually converge the fastest and have the lowest computational cost. However, they also generate the largest amount of messages, with the dynamic approach

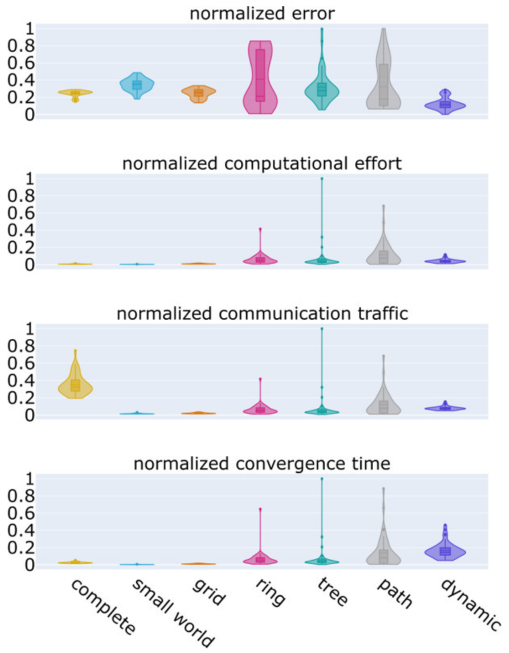
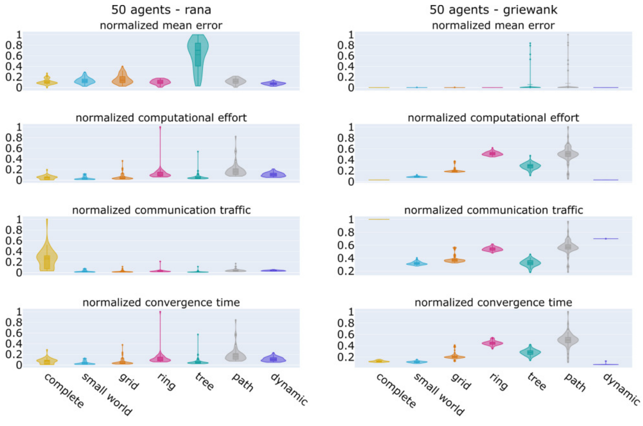


Fig. 7: Results for penalized Schwefel 2.26 (100 agents, $dynamic_{\alpha=1}$)

Fig. 8: Performance results for Rana and Griewank (50 agents, $dynamic_{\alpha=1}$)



(a) Rana

(b) Griewank

performing slightly better. Fig. 8b shows these relationships exemplary for the Griewank function.

To sum up the presented results: In our evaluations, easy to explore fitness landscapes, i.e. functions with uni-modal single funnel shapes and a hardly rugged surface, an arbitrary fairly meshed topology leads reliably to the global optimum. Depending on whether lower message volume or lower computational costs are preferred, small world or the dynamic approach with large α are sufficient. For fitness landscapes with multi-modal and multi-funnel shapes, the dynamic approach with a fast reduction schedule shows advantages regarding solution quality with moderate resource consumption.

VII. Conclusion and Outlook

Distributed optimization heuristics are a suitable approach to handle the increased complexity in cyber-physical multi-energy systems. The communication between the distributed entities must be carefully designed to ensure reliable behavior of a heuristic and thus its suitability for applications in critical infrastructures like energy systems. The communication topology defines which entities exchange information and therefore affects solution quality, convergence speed and collaboration costs. A dynamic approach to topology adaptation during runtime has been presented, based on the principles of simulated annealing. This approach was evaluated against several static topologies using a distributed optimization heuristic to optimize a set of well-known benchmark functions. In addition, we conducted a fitness landscape analysis and trained decision trees to derive correlations between problem properties and performance of the communication topologies.

The main findings can be summarized as follows:

- Functions with small ruggedness at micro level can be considered as "simple". Moreover, if they have a very small DM, i.e., a unimodal shape, many topologies will find the global optimum. Otherwise, more meshed topologies are advantageous.
- When low micro ruggedness is combined with high smoothness or very low macro ruggedness, strongly meshed topologies such as the full graph or the dynamic approach with small values for α, i.e., slowly decreasing connectivity, outperform other topologies.
- The dynamic approach with large values for α excels in providing superior solution quality for functions with high dispersion and thus a difficult multi-funnel landscape. With respect to the cost of cooperation, the dynamic approach is competitive for both easy and hard to explore problems.

Regarding the application domain of distributed control in cyber-physical energy systems, search spaces will have to be analyzed in detail. Given the complexity and non-linearity of the resulting system of systems though, high modality and rugged surfaces seem to be characteristic for many use cases with distributed energy resources, controllable loads and storage systems [36]. So far, communication topologies for distributed heuristics have been selected mostly independently from solution space characteristics. Our work is a first step towards a more systematical selection and dynamic adaptation of communication topologies in this regard.

In future work, we plan to further enhance the dynamic approach. One aspect of this is a intelligent selection of the connections to be removed, by analysing the graph-theoretical characteristics of the intermediate topologies. Furthermore, non-monotonic reduction schedules could be beneficial for some problems, i.e., schedules in which the number of edges can also increase again under certain conditions. The starting topology may be varied as well, since starting with complete graphs is not always advantageous. Overall, with all these enhancements, we aim to be able to optimally select these parameters in the future depending on the problem characteristics and prioritization of the performance dimensions. The resulting parametrizable dynamic topology adaptation will be evaluated on real world problems in the energy domain.

Acknowledgment

We would like to express our sincere gratitude to our college Martin Tröschel, who provided us with advice and support, especially in all data science aspects.

References

[1] P. Vrba, V. Mařík, P. Siano, P. Leitão, G. Zhabelova, V. Vyatkin, and T. Strasser, "A Review of Agent and Service-Oriented Concepts Applied to Intelligent Energy Systems," *IEEE Transactions of Industrial Informatics*, vol. 10, no. 3, pp. 1890–1903, 2014.

[2] S. Ramchurn, P. Vytelingum, A. Rogers, and N. Jennings, "Agent-based homeostatic control for green energy in the smart grid," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, 2011. [Online]. Available: http://dx.doi.org/10.1145/1989734.1989739

[3] M. Sonnenschein, C. Hinrichs, A. Nieße, and U. Vogel, "Supporting Renewable Power Supply through Distributed Coordination of Energy Resources," in *ICT Innovations for Sustainability*, 1st ed., L. M. Hilty and B. Aebischer, Eds. Berlin: Springer, Cham, 2015, vol. 1, pp. 387–404.

[4] A. Nieße, M. Tröschel, and M. Sonnenschein, "Designing Dependable and Sustainable Smart Grids – How to Apply Algorithm Engineering to Distributed Control in Power Systems," *Environmental Modelling & Software*, 2013.

[5] J. Hu, A. Saleem, S. You, L. Nordström, M. Lind, and J. Østergaard, "A multi-agent system for distribution grid congestion management with electric vehicles," *Engineering Applications of Artificial Intelligence*, vol. 38, pp. 45–58, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197614002577

[6] J. Lai, X. Lu, A. Monti, and R. W. de Doncker, "Agent-based voltage regulation scheme for active distributed networks under distributed quantized communication," in *IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society*, vol. 1, 2019, pp. 5941–5946.

[7] A. Nieße, N. Ihle, S. Balduin, M. Postina, M. Tröschel, and S. Lehnhoff, "Distributed ledger technology for fully automated congestion management," *Energy Informatics*, vol. 1, no. 1, p. 22, 2018. [Online]. Available: https://doi.org/10.1186/s42162-018-0033-3

[8] K. Kok, C. Warmer, R. Kamphuis, P. Mellstrand, and R. Gustavsson, "Distributed Control in the Electricity Infrastructure," in *Proceedings of the International Conference on Future Power Systems*, 2005.

[9] S. Lehnhoff, O. Krause, C. Rehtanz, and H. F. Wedde, "Distributed Autonomous Power Management," *at - Automatisierungstechnik*, vol. 3, pp. 167 – 179, 2011.

[10] D. K. Molzahn, F. Dörfler, H. Sandberg, S. H. Low, S. Chakrabarti, R. Baldick, and J. Lavaei, "A survey of distributed optimization and control algorithms for electric power systems," *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 2941–2962, 2017.

[11] Y. Rizk, M. Awad, and E. W. Tunstel, "Decision making in multiagent systems: A survey," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 10, no. 3, pp. 514–529, 2018.

[12] S. Holly and A. Nieße, "On the effects of communication topologies on the performance of distributed optimization heuristics in smart grids," in *INFORMATIK 2020*, R. H. Reussner, A. Koziolek, and R. Heinrich, Eds. Gesellschaft für Informatik, Bonn, 2021, pp. 783–794.

[13] E.-G. Talbi, *Metaheuristics: From Design to Implementation*. John Wiley & Sons, 2009, vol. 74.

[14] M. Ruciński, D. Izzo, and F. Biscani, "On the impact of the migration topology on the island model," *Parallel Computing*, vol. 36, no. 10-11, pp. 555–571, 2010.

[15] T. Crainic, "Parallel metaheuristics and cooperative search," in *Handbook of Metaheuristics*. Springer, 2019, pp. 419–451.

[16] M. Hijaze and D. Corne, "An investigation of topologies and migration schemes for asynchronous distributed evolutionary algorithms," in *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*. IEEE, 2009, pp. 636–641.

[17] ——, "Distributed evolutionary algorithm topologies with adaptive migration schemes," in *2011 IEEE Congress of Evolutionary Computation (CEC)*. IEEE, 2011, pp. 608–615.

[18] M. Sanu and G. Jeyakumar, "Empirical performance analysis of distributed differential evolution for varying migration topologies," *International Journal of Applied Engineering Research*, vol. 10, no. 5, pp. 11–919, 2015.

[19] T.-C. Wang, C.-Y. Lin, R.-T. Liaw, and C.-K. Ting, "Empirical analysis of island model on large scale global optimization," in *2019 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2019, pp. 342–349.

[20] J. Momin and X.-S. Yang, "A literature survey of benchmark functions for global optimization problems," *Int. Journal of Mathematical Modelling and Numerical Optimisation*, vol. 4, no. 2, pp. 150–194, 2013.

[21] X. Li, K. Tang, M. N. Omidvar, Z. Yang, K. Qin, and H. China, "Benchmark functions for the cec 2013 special session and competition on large-scale global optimization," *gene*, vol. 7, no. 33, p. 8, 2013.

[22] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM computing surveys (CSUR)*, vol. 35, no. 3, pp. 268–308, 2003.

[23] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE transactions on evolutionary computation*, vol. 1, no. 1, pp. 67–82, 1997.

[24] Y. Sun, S. K. Halgamuge, M. Kirley, and M. A. Munoz, "On the selection of fitness landscape analysis metrics for continuous optimization problems," in *7th International Conference on Information and Automation for Sustainability*. IEEE, 2014, pp. 1–6.

[25] K. M. Malan and A. P. Engelbrecht, "Quantifying ruggedness of continuous landscapes using entropy," in *2009 IEEE Congress on evolutionary computation*. IEEE, 2009, pp. 1440–1447.

[26] V. K. Vassilev, T. C. Fogarty, and J. F. Miller, "Information characteristics and the structure of landscapes," *Evolutionary computation*, vol. 8, no. 1, pp. 31–60, 2000.

[27] ——, "Smoothness, ruggedness and neutrality of fitness landscapes: from theory to application," *Advances in Evolutionary Computing: Theory and Applications*, p. 3, 2002.

[28] K. M. Malan and A. P. Engelbrecht, "Ruggedness, funnels and gradients in fitness landscapes and the effect on pso performance," in *2013 IEEE Congress on Evolutionary Computation*. IEEE, 2013, pp. 963–970.

[29] M. Lunacek and D. Whitley, "The dispersion metric and the cma evolution strategy," in *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, 2006, pp. 477–484.

[30] M. Locatelli, "A note on the griewank test function," *Journal of global optimization*, vol. 25, no. 2, pp. 169–174, 2003.

[31] C. Hinrichs and M. Sonnenschein, "A distributed combinatorial optimisation heuristic for the scheduling of energy resources represented by self-interested agents." *IJBIC*, vol. 10, no. 2, pp. 69–78, 2017.

[32] J. Bremer and S. Lehnhoff, "An agent-based approach to decentralized global optimization-adapting cohda to coordinate descent," in *International Conference on Agents and Artificial Intelligence*, vol. 2. SCITEPRESS, 2017, pp. 129–136.

[33] H.-G. Beyer and S. Finck, "Happycat–a simple function class where well-known direct search algorithms do fail," in *International conference on parallel problem solving from nature*. Springer, 2012, pp. 367–376.

[34] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*. CRC press, 1984.

[35] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[36] A. Nieße, J. Bremer, and S. Lehnhoff, "On local minima in distributed energy scheduling." in *FedCSIS (Position Papers)*, 2017, pp. 61–68.

DOI: 10.15439/2021F36
ISSN 2300-5963 ACSIS, Vol. 25

Minimizing Tardiness in a Scheduling Environment with Jobs' Hierarchy

Michal Sinai and Tami Tamir
School of Computer Science
The Interdisciplinary Center
Herzliya, Israel
Emails: michal.sinai@post.idc.ac.il, tami@idc.ac.il

***Abstract*—In many scheduling environments, some jobs have higher priority than others. Such scenarios are theoretically modelled by associating jobs with weights, or by having precedence constraints that limit jobs' processing order. In this paper we define and consider a new model, motivated by real-life behaviour, in which the priority among jobs is defined by a *dominance hierarchy*. Specifically, the jobs are arranged in hierarchy levels, and high ranking jobs are ready to accept only outcomes in which the service they receive is better than the service of subordinate jobs. We first define the model and the set of feasible schedules formally. We then consider two classical problems: minimizing the maximal tardiness and minimizing the number of tardy jobs. We provide optimal algorithms or hardness proofs for these problems, distinguishing between a global objective function and a multi-criteria objective.**

I. Introduction

JOB Scheduling problems are considered to be a fundamental and well studied field in theoretical computer science. The study of the combinatorial optimization problems induced by various scheduling environments is motivated by numerous real-life applications arising in production planning, traffic control, cloud computing services, and many more. A typical scheduling problem instance involves assigning a set of n independent jobs on m parallel machines in a way that optimally utilizes the machines and achieves high quality of service for the jobs. These objectives are mathematically modelled by minimizing a predefined objective function, such as the makespan (maximal completion time of some job), total completion time, lateness, etc. We refer to [18] for a comprehensive survey of various models of scheduling problems.

In many scheduling environments, the jobs are not treated in a fair way. Naturally, some jobs have higher priority than others. Such scenarios are theoretically modelled in two ways: (i) jobs are associated with weights that reflect their priority. The jobs' performance measure is scaled by the weight, thus jobs with higher weight get better quality of service. (ii) the scheduling instance includes a directed acyclic graph describing precedence constraints among jobs. A directed edge from job j_1 to job j_2 implies that the processing of j_2 can start only after the processing of j_1 is completed.

In this paper we study a scheduling setting, motivated by real-life behaviour, in which the priority among jobs is defined in a different way. Our model reflects real-life environments in which the schedule is not determined completely by the system. Traditionally, scheduling problems have been studied from a centralized point of view, that is, a centralized authority, 'the scheduler' determines the assignment. Many modern systems provide service to multiple strategic users, who may influence the possible outcomes. As a result, non-cooperative game theory has become an essential tool in the analysis of job-scheduling applications [21], [5]. Our model studies a natural setting, in which the users are arranged in a *dominance hierarchy*, and high ranking users are ready to accept only outcomes in which the service they receive is better than the service of subordinate users.

In behavioral sciences, the study of dominance hierarchy is based on the fact that different organisms have different aggressiveness levels. Aggression is defined as a behavior which is intended to increase the social dominance of the organism relative to the dominance position of other organisms [6]. Different levels of aggressiveness lead to a dominance hierarchy - a type of social hierarchy that arises when members of a social group interact [4], [9]. Highly rank members of the society have better access to valuable resources such as mates and food. Our model is inspired by such environments. Specifically, in our setting, the jobs are partitioned into c hierarchical levels. High-ranking jobs can bypass subordinate jobs if this improves their performance. Moreover, all the jobs, from all hierarchy levels cooperate and are ready to modify their assignment if this modification does not harm their performance, and may help other high-ranking jobs get an advantage over subordinate ones.

In Section II we define the model and the set of feasible schedules formally. We also present algorithms for testing the feasibility of a schedule with respect to jobs' tardiness and with respect to the lateness indicator. In Section III we present optimal algorithms for the problem of minimizing the maximal tardiness of a job. In Section IV we consider the problem of minimizing the number of tardy jobs. We distinguish between the global objective in which the goal is to find a feasible schedule that minimizes the total number of tardy jobs, independent of their hierarchy level, and the multi-criteria objective, in which the primary goal is to minimize the number of tardy highly ranked jobs, the secondary goal is to minimize the number of tardy jobs from the 2nd rank, and so on. For a constant number of hierarchy levels we present an optimal algorithm for the multi-criteria objective, and an NP-hardness proof for the global objective.

II. Preliminaries

Let $\mathcal{J}$ be a set of jobs. Every job $J_j \in \mathcal{J}$ has a processing time p_j, as well as a due-date d_j, denoting the time in which it should be completed. The set $\mathcal{J}$ consists of c sets; $\mathcal{J} = \mathcal{J}_1 \cup \mathcal{J}_2 \cup \ldots \cup \mathcal{J}_c$, where $\mathcal{J}_k$ is a set of jobs in the ℓ-th hierarchy level. That is, the jobs of $\mathcal{J}_1$ have the highest rank, and the jobs of $\mathcal{J}_c$ are the most subordinate. Let $n = \sum_{\ell=1}^{c} |\mathcal{J}_\ell|$. A schedule π on a single machine determines a non-preemptive assignment of the jobs on the machine. For a schedule π and a job J_j, let $S_j(\pi), C_j(\pi)$ denote the start time and the completion time of J_j in π. In our setting, all the jobs are available at time 0 (no release times), and no preemptions are allowed, therefore, for every job j, $C_j(\pi) - S_j(\pi) = p_j$. Also, w.l.o.g., we only consider schedules with no intended idle. Clearly, idle segments can be removed by shifting some jobs to start earlier. By the above, a schedule π can be described by specifying an order of the jobs, and $C_j(\pi) = \sum_{j':S_{j'}(\pi) \leq S_j(\pi)} p_{j'}$.

For a given schedule π, let $L_j(\pi) = C_j(\pi) - d_j$ denote the lateness of job J_j in π. The jobs need to be ready by their due-date; early completion of a job has no effect on the quality of service, thus, the study of scheduling environments in which jobs are associated with due-dates, considers mostly the two following measurements:

1) $T_j(\pi) = \max\{0, C_j(\pi) - d_j\}$ is the *tardiness* of job J_j.
2) $U_j(\pi) \in \{0, 1\}$ is a binary *lateness indicator* indicating whether J_j is *tardy*, that is, $U_j(\pi) = 1$ if and only if $C_j(\pi) > d_j$.

For a set $\mathcal{J}_\ell$, let $T_{\mathcal{J}_\ell}(\pi) = \max_{j \in \mathcal{J}_\ell} T_j(\pi)$ be the maximal tardiness of a job in $\mathcal{J}_\ell$. For the lateness indicator we measure the performance of a set of jobs by the number of tardy jobs in the set, in particular, $U_{\mathcal{J}_\ell}(\pi) = \sum_{j \in \mathcal{J}_\ell} U_j(\pi)$ is the number of tardy jobs in $\mathcal{J}_\ell$.

We will analyze two objective functions. The first is minimizing the maximal tardiness, and the second is minimizing the number of tardy jobs. Using the common three-fields notation for theoretic scheduling problems [10], we denote the corresponding problems in the presence of hierarchy levels by $1|hierarchy|T_{max}$ and $1|hierarchy|\sum U_j$.

High rank jobs can bypass and push subordinate jobs. They also cooperate with each other. Formally, a schedule π is considered *feasible* if for every hierarchy level $1 \leq \ell \leq c$, and every job $J_i \in \mathcal{J}_\ell$ it holds that J_i cannot improve its objective value by bypassing less dominant jobs, even if all the jobs having rank at least ℓ are ready to modify their assignment as long as they are not harmed. This general definition has a different practical meaning depending on the objective function. Specifically:

Definition 2.1: A schedule π is *feasible with respect to tardiness* if for for every rank $1 \leq \ell \leq c$ and every tardy job $J_i \in \mathcal{J}_\ell$ it holds that there is no schedule π' such that $C_i(\pi') < C_i(\pi)$ and for every job $J_j \in \cup_{1 \leq k \leq \ell} \mathcal{J}_k$ it holds that $T_j(\pi') \leq T_j(\pi)$. In other words, there is not schedule in which J_i has a reduced tardiness, and no job from a higher or equal hierarchy level has a higher tardiness.

Definition 2.2: A schedule π is *feasible with respect to the number of tardy jobs* if for every rank $1 \leq \ell \leq c$ and every tardy job $J_i \in \mathcal{J}_\ell$ it holds that there is no schedule π' such that $C_i(\pi_i) \leq d_i$ and for every job $J_j \in \cup_{1 \leq k \leq \ell} \mathcal{J}_k$ it holds that $U_j(\pi') \leq U_j(\pi)$. Thus, J_i completes on time and if a same or higher hank job J_j is not tardy in π it must complete in time also in π'.

Note that if the objective of a job is merely to minimize its completion time, then the hierarchy induces an order according to which jobs of different levels must be processed, and finding an optimal solution on a single machines is an easy task. Objective functions that depend on jobs' tardiness are more challenging since a job may have a high completion time and still perform perfectly as long as it is not tardy. Thus, the order of jobs in an optimal schedule does not necessarily agree with their ranks. This observation is crucial in understanding the model and the involved challenges.

The general problem we consider is finding a feasible schedule that optimizes the objective function, that is, minimize the maximal tardiness of a job, or minimizes the number of tardy jobs. A different goal that we consider is a multi-criteria one. Specifically, the primary goal is to optimize the schedule for $\mathcal{J}_1$. Out of all feasible schedules achieving the best for $\mathcal{J}_1$, the goal is to optimize the schedule for the jobs in $\mathcal{J}_2$, and so on. We use the notation $1|hierarchy|(\gamma_1, \ldots, \gamma_c)$ the denote the problem with the multi-criteria objective function γ. E.g., for $c = 2$, in the problem $1|hierarchy|(U_\mathcal{A}, U_\mathcal{B})$, the primary goal is to minimize the number of tardy dominant jobs, and among all the feasible schedules achieving this objective, minimize the number of tardy subordinate jobs.

We conclude the introduction with an example that demonstrates the optimality with respect to the general and the multi-criteria objective function. Consider the problem of minimizing the number of late jobs. That is, $1|hierarchy|\sum U_j$. Assume $c = 2$. Let $\mathcal{A} = \{a_1, a_2, a_3\}$ be the set of dominant jobs, where $p_1 = p_2 = L$ and $p_3 = L + 1$, for some constant $L > 2$. The set $\mathcal{B}$ of subordinate jobs includes $L-1$ unit-length jobs. Note that $n = |\mathcal{A}| + |\mathcal{B}| = L + 2$. Assume further that all the jobs in the instance have the same due-date $d_j = 2L$. An optimal schedule for $\sum U_j$ is the schedule π_1, presented in the top of Figure 1. There are 2 tardy jobs. The longer dominant job, a_3, and $L - 1$ subordinate jobs complete on time. The schedule π_1 is feasible, even though a_1 and a_2 are late. None of these jobs can benefit from bypassing subordinate jobs, as their total processing time is less than L. The schedule π_2 in Figure 1 is optimal for the problem $1|hierarchy|(U_\mathcal{A}, U_\mathcal{B})$. The two dominant jobs a_1 and a_2 are not late, and the other L jobs are late. The above example illustrates some of the challenges in scheduling jobs with different hierarchy levels, and the difference from the global objective function and the multi-criteria one.

Related work: Job scheduling on a single machine has been widely studied. When there are no precedence constraints or

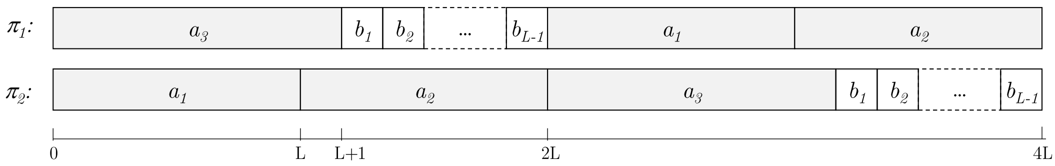


Fig. 1. π_1 is optimal for $1|hierarchy|\sum U_j$, while π_2 is optimal for $1|hierarchy|(U_1, U_1)$.

weights, the problem $1||T_{max}$ is solved optimally by Earliest Due-Date first (EDD) rule, that schedule the jobs in non-decreasing order of due-date [15]. The problem $1||\sum U_j$ is solved by Moore's algorithm [17].

When jobs are associated with weights, the problem $1||\sum w_j Uj$ becomes NP-hard even when the jobs all have common due-date [13]. Pseudo-polynomial time algorithms are given in [16] and [19]. If the number of different job weights is a constant, then $1||\sum w_j Uj$ is solvable in polynomial time [11]. The unweighted problem of minimizing the number of tardy jobs is strongly NP-hard when the jobs' processing order must obey some precedence constraints. This is true even if the precedence constraints are limited to chains and all jobs have unit length, that is, $1|chains; p_j = 1|\sum U_j$. See [1] for a survey on algorithms for single machine scheduling to minimize weighted number of tardy jobs.

For the maximum tardiness problem, the addition of weights does not change the complexity of the problem, that is, the weighted problem, $1|| \max_j w_j T_j$, is solvable in polynomial time [12], [7]. Moreover, the problem remains tractable even in the addition of arbitrary precedence constraints [15]. On the other hand, when a machine can process several jobs simultaneously (batch-scheduling), the problem becomes NP-hard [3].

Aggressiveness is a lighter notion of priority. Dominant jobs can be processed after less dominant ones, if their performance is not harmed. An environment in which some jobs are aggressive is studied in [20], where a new notion of selfish precedence constraint is defined. The paper presents algorithms for scheduling jobs on parallel machines, where some of the jobs are aggressive. An aggressive job do not let non-aggressive jobs start processing before it. Additional relaxed models of precedence constraints are studied in [14], [2].

A. Feasibility Tests

In this section we present algorithms for testing whether a given schedule is feasible with respect to some objective. For a schedule π, the algorithms returns *True* if π is feasible, or *False due to* J_i, if π is not feasible since some job J_i can benefit from rearranging the jobs.

Algorithm 1 performs a feasibility test of a given schedule with respect to the jobs' tardiness. It proceeds by verifying, for every tardy job J_i, that there is no schedule in which J_i has a reduced tardiness and the non-tardy jobs from hierarchy levels at least as high as J_i are not harmed, as required by Definition 2.1.

Algorithm 1 - Feasibility test of a schedule π w.r.t T_i

```
1: Let $\mathcal{J}_{tardy}$ and $\mathcal{J}_{in.time}$ be, respectively, the set of tardy and non-tardy jobs in $\pi$.
2: for each job $J_i \in \mathcal{J}_{tardy}$ do
3:    Assume $J_i \in \mathcal{J}_\ell$.
4:    Let $S_1 = \cup_{1 \le k \le \ell} \mathcal{J}_k \cap \mathcal{J}_{tardy}$.
5:    Let $S_2 = \cup_{1 \le k \le \ell} \mathcal{J}_k \cap \mathcal{J}_{in.time}$.
6:    Let $\pi'_i$ be a schedule of $S_1 \cup S_2 \setminus \{J_i\}$ produced in the following way:
7:       Assign the jobs in $S_1 \setminus \{J_i\}$ as in $\pi$.
8:       Add the jobs in $S_2$ in non-increasing order of due-date. Every job $J_j$ is assigned, possibly with preemptions, in the latest available slots in $[0, d_j]$.
9:    If $\pi'_i$ includes more than $p_i$ idle slots in $[0, C_i(\pi)]$ then return False due to $J_i$.
10: end for
11: return True.
```

Lemma 2.1: Algorithm 1 returns *True* if and only if π is feasible with respect to T_i.

Proof: The algorithm proceeds by checking feasibility for every tardy job separately. Clearly, if a job J_i is not late, then the schedule is feasible for it. If J_i is late, then S_1 and S_2 are the sets of tardy and non-tardy jobs that are ranked in the hierarchy at least as high as J_i. We check whether there exists a schedule in which these jobs are not harmed, and the tardiness of J_i is reduced.

Assume that the algorithm returns *False due to* J_i. Assume $J_i \in \mathcal{J}_\ell$. We show that there exists a schedule π' such that $C_i(\pi') < C_i(\pi)$ and for every job $J_j \in \cup_{1 \le k \le \ell} \mathcal{J}_k$ it holds that $T_j(\pi') \le T_j(\pi)$.

The schedule π' is produced from the schedule π'_i build in steps 6–8. First, preemptions are removed: if job J_j is preempted in π'_i, then in π' it is processed non preemptively in $[C_j(\pi'_i) - p_j, C_j(\pi'_i)]$. Jobs that were processed in this interval are shifted to start earlier. The tardiness of J_j does not change, as its completion time remains $C_j(\pi'_i)$. The tardiness of the shifted job could only decrease. After the preemption removal, we add J_i in the earliest idle slots, possibly with preemptions. Since the condition in step 9 is met, $T_i(\pi') < T_i(\pi)$. Next, if J_i is scheduled with preemptions, then preemptions are removed, without harming any of the completion times, as described above. Finally, the jobs from lower hierarchy levels $\cup_{\ell < k \le c} \mathcal{J}_k$ are added in arbitrary way.

Note that it is always possible to add the jobs of S_2 as required in Step 8 of the algorithm, since they are not late in π, thus, there is clearly sufficient space for them on the machine when the jobs of $\cup_{\ell<k\leq c}\mathcal{J}_k$ are removed.

By the condition in Step 9, the lateness of J_i in π' is lower than its lateness in π. Since all other jobs with at least the same rank are not harmed, we get a contradiction to the stability of π.

Assume that the algorithm returns *True*. It means that for every tardy job j_i in π, there are at most p_i idle slots in $[0, C_i(\pi)]$ in a schedule in which jobs of lower rank are removed, and each of the remaining jobs is scheduled as late as possible. This implies that it is not possible to rearrange the jobs in π such that J_i reduces its tardiness, without harming the performance of at least one job with rank higher or equal to rank of J_i. Thus, π is feasible. ■

We turn to consider objectives that refer to the lateness indicator. Algorithm 2 performs a feasibility test of a given schedule with respect to the lateness indicator. The algorithm proceeds by verifying, for every tardy job, J_i, that there is no schedule in which J_i completes on time and the jobs from hierarchy levels at least as high as J_i are not harmed, as required by Definition 2.2.

Algorithm 2 - Feasibility test of a schedule π w.r.t U_i

```
1: Let $\mathcal{J}_{tardy}$ and $\mathcal{J}_{in.time}$ be, respectively, the set of tardy
   and non-tardy jobs in $\pi$.
2: for each job $J_i \in \mathcal{J}_{tardy}$ do
3:     Assume $J_i \in \mathcal{J}_\ell$.
4:     Let $S_2 = \cup_{1\leq k\leq\ell}\mathcal{J}_k \cap \mathcal{J}_{in.time}$.
5:     Let $\pi'_i$ be a schedule in EDD order of the jobs in $S_2 \cup$
       $\{J_i\}$.
6:     If no job is late in $\pi'_i$ then return False due to $J_i$.
7: end for
8: return True.
```

Lemma 2.2: Algorithm 2 returns *True* if and only if π is feasible with respect to U_i.

Proof: The problem $1||T_{max}$ is known to be solvable optimally by EDD rule. In particular, if for some instance of $1||T_{max}$, there exists a schedule in which no job is late, that is, $T_{max} = 0$, then no job is late if the jobs are processed in EDD order. Algorithm 2 is based on the above fact.

Assume that the algorithm returns *False*. This implies that for some late job, there exists a schedule of $S_2 \cup \{J_i\}$ in which no job is late. Thus, π can be replaced by the schedule π'_i built in step 5, followed by a schedule in arbitrary order of the jobs that are late in π. This modified schedule is better for J_i and does not harm the objective value of any job in $\cup_{1\leq k\leq\ell}\mathcal{J}_k$, as required. Thus, π is not feasible.

Assume that the algorithm returns *True*. It means that for every late job in π, at least one job would be late in a schedule in which the jobs of $S_2 \cup \{J_i\}$ are processed in EDD order. Since EDD is optimal for $1||T_{max}$, there is no schedule in which none of these jobs is late. This implies that it is not possible to rearrange the jobs in π such that J_i is not late, without harming the performance of at least one job with rank at least as high as J_i. Thus, π is feasible. ■

III. Minimizing Maximal Tardiness

A. The multi-criteria objective function: $1|hierarchy|(T_{\mathcal{J}_1}, \ldots, T_{\mathcal{J}_c})$

In this section we consider the multi-criteria objective function of minimizing the maximal tardiness. Formally, recall that for every $1 \leq \ell \leq c$, $T_{\mathcal{J}_\ell}(\pi) = \max_{j\in\mathcal{J}_\ell} T_j(\pi)$ denotes the maximal tardiness of a job in $\mathcal{J}_\ell$ in a schedule π. An optimal schedule achieves the minimal possible $T_{\mathcal{J}_1}$ and for all $\ell > 1$ it achieves the minimal possible $T_{\mathcal{J}_\ell}$ among all schedules that achieve the minimal $T_{\mathcal{J}_k}$ for every $1 \leq k < \ell$.

We present an optimal algorithm for the problem. Recall that algorithm EDD, that schedule the jobs in non-decreasing due-date order is optimal for the problem when there are no hierarchy levels. The algorithm is presented for $c = 2$, that is, $\mathcal{J} = \mathcal{A} \cup \mathcal{B}$, where $\mathcal{A}$ is a set of dominant jobs, and $\mathcal{B}$ a set of subordinate jobs. At the end of this section we explain how to generalize it for $c > 2$ hierarchy levels.

Algorithm 3 constructs an optimal schedule π in two phases. First, all the jobs are assigned according to EDD order, and then the schedule is turned into a feasible one, by letting some dominant jobs pass some subordinate jobs. Recall that for a schedule π and a job J_i, we denote by $S_i(\pi)$ and $C_i(\pi)$ the start time and the completion time of J_i in π.

Algorithm 3 - An optimal algorithm for $1|hierarchy|(T_\mathcal{A}, T_\mathcal{B})$

```
1: Schedule all jobs according to EDD order, that is, $d_1 \leq$
   $d_2 \leq \cdots \leq d_n$.
2: Let $\pi$ be the schedule produced by EDD.
3: for each job $J_i \in \mathcal{A}$ according to their order in $\pi$ do
4:     while $J_i$ is late and at least one job from $\mathcal{B}$ precedes it
       do
5:         Let $J_k$ be the job in $\mathcal{B}$ for which $S_k(\pi) < S_i(\pi)$,
           and $S_k(\pi)$ is maximal.
6:         Shift the jobs scheduled in $[C_k(\pi), C_i(\pi)]$ earlier by
           $p_k$ units.
7:         Schedule $J_k$ right after $J_i$.
8:     end while
9: end for
```

Theorem 3.1: Algorithm 3 produces a feasible schedule, optimal for the bi-criteria problem $(T_\mathcal{A}, T_\mathcal{B})$.

Proof: Let π be the schedule produced by the algorithm for an input $\mathcal{A} \cup \mathcal{B}$. Every dominant job $J_i \in \mathcal{A}$ is considered in the while loop. Note that in the shifts performed in step 6, the jobs that are shifted forward are all dominant, since J_k is the last $\mathcal{B}$-job before J_i. Combining this with the initial EDD order, we get,

Observation 3.2: In π, the jobs in $\mathcal{A}$ are processed in EDD order, and the jobs in $\mathcal{B}$ are processed in EDD order.

The while loop terminates if J_i is not late or if it is preceded only by $\mathcal{A}$-jobs with lower or equal due-date. Also, by Observation 3.2, in the final schedule, every $\mathcal{B}$-job is preceded by $\mathcal{B}$-jobs with lower or equal due-date, or $\mathcal{A}$-jobs that bypassed it in order to reduce their tardiness. Thus, π is feasible.

Next note that no $\mathcal{A}$-job that is processed after a $\mathcal{B}$-job is late. Thus, as illustrated in Figure 2, the schedule π begins with a sequence of $\mathcal{A}$-jobs that are processed in a row, and are possibly late, followed a mixture of $\mathcal{B}$-jobs and non-late $\mathcal{A}$-jobs. Let J_z be the last late $\mathcal{A}$-job in π. Let $\mathcal{A}_1$ be the subset of $\mathcal{A}$-jobs that are processed sequentially in $[0, C_z(\pi)]$, and let $\mathcal{A}_2$ be the set of remaining $\mathcal{A}$-jobs, that are not late and are processed interleaved with $\mathcal{B}$-jobs after $C_z(\pi)$. We prove the optimality of π by considering separately the prefix in which the jobs of $\mathcal{A}_1$ are processed and the suffix in which the jobs of $\mathcal{A}_2 \cup B$ are processed.

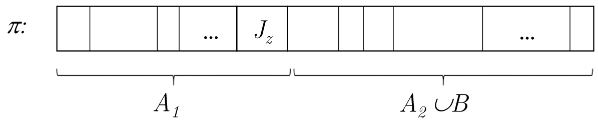


Fig. 2. The structure of the optimal schedule π

Lemma 3.3: Every optimal schedule $\pi^\star$ can be modified such that it agrees with π on the assignment of $\mathcal{A}_1$, without harming its feasibility nor the objective function.

Proof: Since J_z is the last late $\mathcal{A}$-job in π, which is a feasible schedule, any solution that schedules some $\mathcal{B}$-job, J_b, in the interval $[0, C_z(\pi)]$ is not feasible, as J_z may reduce its tardiness by bypassing J_b. Thus, in any feasible schedule, only $\mathcal{A}$-jobs are processed in the interval $[0, C_z(\pi)]$.

Assume that π^* does not agree with π on the assignment of $\mathcal{A}_1$, and let $J_i \in \mathcal{A}_1$ be the first job in π that has a higher starting time in $\pi^\star$. We use an exchange argument to show that π^* can be converted to agree with π on the assignment of J_i without harming its feasibility nor increasing the maximal tardiness of jobs in $\mathcal{A}$ or $\mathcal{B}$. Let H be the set of jobs that are scheduled in $\pi^\star$ during the interval $[S_i(\pi), S_i(\pi^\star)]$. Let π' be the schedule obtained from $\pi^\star$ by moving J_i before H in $\pi^\star$ (see Figure 3). By Observation 3.2, each of these jobs has higher due-date than d_i.

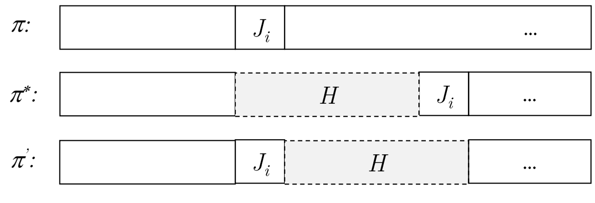


Fig. 3. Converting $\pi^\star$ to a profile π' that agrees with π on the assignment of job $J_i \in \mathcal{A}$.

The schedule π' is feasible: By the feasibility of π^*, the set H includes only $\mathcal{A}$-jobs, as otherwise, J_i or another job from $\mathcal{A}_1$ is tardy and can reduce its tardiness by bypassing the $\mathcal{B}$-jobs in H.

Since in π, the jobs of $\mathcal{A}_1$ are processed in EDD order, for every $J_k \in H$, it holds that $d_i \leq d_k$ and $C_k(\pi') \leq C_i(\pi^\star)$. Therefore, the lateness of J_k in π' is not higher than the lateness of J_i in $\pi^\star$, and the maximal tardiness among the jobs in $\mathcal{A}$ is not harmed. The jobs that are processed after $C_i(\pi^\star)$ are not affected by the exchange, and their tardiness does not change.

By repeating the above exchange argument as long as $\pi^\star$ does not agree with π on the assignment of $\mathcal{A}_1$, we get the statement of the lemma. ∎

We turn to consider the jobs of $\mathcal{B} \cup \mathcal{A}_2$. These jobs are processed after time $C_z(\pi)$.

Claim 3.4: Every optimal schedule $\pi^\star$ can be modified such that no job in $\mathcal{A}_2$ is late, without delaying any job in $\mathcal{B}$.

Proof: Let $J_i \in \mathcal{A}_2$ be a late job in $\pi^\star$. Since $\pi^\star$ is feasible, J_i is precedes only by $\mathcal{A}$-jobs. Also, we can assume that the machine is not idle between these jobs, as otherwise, idles can be removed by shifting the jobs to start earlier, without harming the feasibility or the quality of the solution. Modify $\pi^\star$ be rearranging in EDD order J_i and the $\mathcal{A}$-jobs from $\mathcal{A}_2$ that precedes it. From the optimally of EDD, the maximal tardiness of the $\mathcal{A}$-jobs in the resulting schedule is equal to or lower than their maximal tardiness before the modification. After the reorder, the order of the jobs agrees with π, and since in π no job from $\mathcal{A}_2$ is late, this is true for $\pi^\star$ as well. ∎

Based on Claim 3.4, we can assume w.l.o.g., that no jobs in $\mathcal{A}_2$ is late in $\pi^\star$.

Lemma 3.5: Every optimal schedule $\pi^\star$ can be modified such that it agrees with the assignment of $\mathcal{A}_2 \cup \mathcal{B}$ in π without harming its feasibility nor the objective function.

Proof: We show that $\pi^\star$ can be converted to agree with π. Specifically, we use an exchange argument for handling the leftmost disagreement. The same argument can be applied as long as the schedules are not identical.

Let $J_i \in \mathcal{A}_2 \cup \mathcal{B}$ be the first job in π that has a different starting time in $\pi^\star$. Let H be the set of jobs that are scheduled in $\pi^\star$ during the interval $[S_i(\pi), S_i(\pi^\star)]$. We distinguish between two cases depending on the hierarchy level of J_i.

Assume first that $J_i \in \mathcal{A}_2$. As in the case $J_i \in \mathcal{A}_1$, let π' be the schedule obtained from $\pi^\star$ by moving i before H in $\pi^\star$ (see Figure 3). We show that π' is feasible and has the same objective value. Consider an $\mathcal{A}$-job $J_k \in H$. Since Algorithm 3 schedules $\mathcal{A}$-jobs by EDD order, for every $\mathcal{A}$-job $J_k \in H$, it holds that $d_k \geq d_i$. By Claim 3.4, J_i is not late in $\pi^\star$, hence $C_i(\pi^\star) \leq d_i$. For every $\mathcal{A}$-job $J_k \in H$, we have that $C_k(\pi') \leq C_i(\pi^\star) \leq d_i \leq d_k$, that is, J_k is not late in π'.

We conclude that all $\mathcal{A}$-jobs in H will not be late after the modification, therefore the maximal tardiness of the $\mathcal{A}$-jobs is not affected.

Consider now a $\mathcal{B}$-job $J_k \in H$. Algorithm 3 schedules J_i before J_k in two cases:

1) $d_i \leq d_k$. By the feasibility of π^*, J_i is not late in $\pi^\star$. Since $d_i \leq d_k$, J_k is not late in $\pi^\star$ as well. In this case, J_k will not be late after the modification, since $C_k(\pi') \leq C_i(\pi^\star) \leq d_i \leq d_k$.
2) $d_i > d_k$. We show that this case never happens. J_i is scheduled in π before J_k even-though $d_i > d_k$ since J_k was delayed when some $J_\ell \in \mathcal{A}$ is considered in the

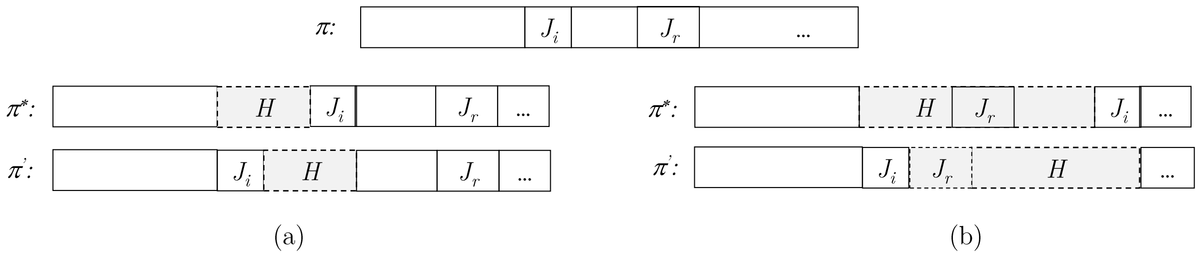


Fig. 4. $J_i \in \mathcal{B}$. (a) J_r is scheduled after J_i in $\pi^\star$, (b) J_r is scheduled before J_i in $\pi^\star$

while loop (possibly $\ell = i$). By the algorithm, $d_i \leq d_\ell$, and a consequent set of jobs from $\mathcal{A}$ are processed in π in EDD order in $[S_i(\pi), C_\ell(\pi)]$. Moreover, since J_k must be delayed in order to prevent J_ℓ from being late, at least one of these jobs will be late if p_k precedes one of them, contradicting the feasibility of $\pi^\star$.

Therefore, the maximal tardiness of the $\mathcal{B}$-jobs in H is not affected by the modification.

We turn to consider the case $J_i \in \mathcal{B}$.

Let H be the set of jobs that are scheduled in $\pi^\star$ during the interval $[S_i(\pi), S_i(\pi^\star)]$, and let J_r be the first $\mathcal{A}$-job scheduled after J_i in π. Note that J_r has the minimal due date among all $\mathcal{A}$-jobs that follow J_i in π. We distinguish between two cases:

1) J_r is scheduled after J_i in $\pi^\star$. Let π' be the schedule obtained from $\pi^\star$ by moving J_i before H in $\pi^\star$ (see Figure 4(a)). We show that π' is feasible and has the same objective value. First, we show that the maximal tardiness of the jobs in $\mathcal{B} \cap H$ in π' is not higher than the maximal tardiness of these jobs in $\pi^\star$.
For J_i, the exchange is clearly beneficial, as it is moved to start earlier. For every other $\mathcal{B}$-job $J_k \in H$, the EDD order applied in Algorithm 3 implies that $d_i \leq d_k$, thus,

$$\begin{aligned} L_k(\pi') = C_k(\pi') - d_k &\leq C_i(\pi^\star) - d_k \\ &\leq C_i(\pi^\star) - d_i = L_i(\pi^\star) \end{aligned}$$

We conclude that

$$max(L_k(\pi'), L_i(\pi')) \leq max(L_k(\pi^\star), L_i(\pi^\star)).$$

Since $T_j = max(L_j, 0)$ we get,

$$max(T_k(\pi'), T_i(\pi')) \leq max(T_k(\pi^\star), T_i(\pi^\star)).$$

Therefore, the maximal tardiness of a job in $\mathcal{B} \cap H$ is not harmed.
Next, we consider the jobs in $\mathcal{A}_2 \cap H$. Since J_r has the minimal due date among all $\mathcal{A}$-jobs that follow J_i in π, every $\mathcal{A}$-job $J_k \in H$ satisfies $d_k \geq d_r$. In addition, by Claim 3.4, J_r is not late in $\pi^\star$. Thus,

$$C_k(\pi') \leq C_i(\pi^\star) \leq C_r(\pi^\star) \leq d_r \leq d_k.$$

Therefore, no $\mathcal{A}$-job in $\mathcal{A}_2 \cap H$ is late in π', and the maximal tardiness is not affected.

2) J_r is scheduled before J_i in $\pi^\star$ (see Figure 4(b)). In this case, let π' be the schedule obtained by moving J_i to precede H in $\pi^\star$, and reorder the jobs in H, such that $\mathcal{A}$-jobs in H appear first, in EDD order, and are followed by the $\mathcal{B}$-jobs in H, also in EDD order. Note that $J_r \in H$ and since it has the minimal due date among the $\mathcal{A}$-jobs that follow J_i in π, it is now processed right after J_i.
Clearly, J_i is not late in π'. J_r is not late in $\pi^\star$ and in π. Since it is processed after J_i in π, it is not late in π' either. The remaining $\mathcal{A}$-jobs in H are processed in both π and π' in EDD order and are preceded by both J_i and J_r. Since they are not late in π, they are not late in π' either. The $\mathcal{B}$-jobs in H are processed last, in EDD order. For each such job J_k, by Algorithm 3, $d_i \leq d_k$, therefore, $L_k(\pi') = C_k(\pi') - d_k \leq C_i(\pi^\star) - d_i = L_i(\pi^\star)$. We conclude that the maximal tardiness of the $\mathcal{B}$-jobs in π' is not be higher than the tardiness of J_i in $\pi^\star$. Therefore, the modification does not increase the maximal tardiness of a job in $\mathcal{B}$.

By repeating the above exchange argument, as long as $\pi^\star$ does not agree with π, we conclude that every optimal schedule can be modified such that it agrees with π, and its objective value is not harmed. Also, the initial EDD order implies the feasibility for the $\mathcal{B}$-jobs. That is, no $\mathcal{B}$-job can benefit from rearranging other jobs in a way that reduces its tardiness and does not harm any of the other jobs. ■

Combining Lemmas 3.3 and 3.5, we conclude that Algorithm 3 produces a feasible schedule that is optimal for the bi-criteria objective $(T_\mathcal{A}, T_\mathcal{B})$. ■

Extension for $c > 2$ hierarchy levels: Algorithm 4 extends Algorithm 3 for more than two hierarchy levels. The idea is to consider the levels one after the other. When the set $\mathcal{J}_\ell$ is considered, all the sets $J_\ell + 1, \ldots, \mathcal{J}_c$, can be viewed as a single subordinate level, and is therefore treated as the class $\mathcal{B}$ in the case of $c = 2$.

Algorithm 4 - An optimal algorithm for $1|hierarchy|(T_{\mathcal{J}_1},\dots,T_{\mathcal{J}_c})$

```
Schedule all jobs according to EDD order, that is, $d_1 \le d_2 \le \cdots \le d_n$.
Let $\pi$ be the schedule produced by EDD.
for $\ell = 1$ to $c-1$ do
    Let $\mathcal{B} = \cup_{j=\ell+1}^{c} \mathcal{J}_j$
    for each job $J_i \in \mathcal{J}_\ell$ according to their order in $\pi$ do
        while $J_i$ is late and at least one job from $\mathcal{B}$ precedes it do
            Let $J_k$ be the job in $\mathcal{B}$ for which $S_k(\pi) < S_i(\pi)$, and $S_k(\pi)$ is maximal.
            Shift the jobs scheduled in $[C_k(\pi), C_i(\pi)]$ earlier by $p_k$ units.
            Schedule $J_k$ right after $J_i$.
        end while
    end for
end for
```

The proof of the algorithm follows the structure of the proof of Algorithm 3. We show by induction that for every $1 \le \ell < c$, the schedule after ℓ iterations is optimal with respect to the multi-criteria objective of the ℓ high hierarchy levels. The initial EDD order implies the optimality and feasibility for the subordinate class, $\mathcal{J}_c$.

B. The global objective function: $1|hierarchy|T_{max}$

We turn to consider the global objective function of minimizing the maximal tardiness of a job. Unlike the multi-criteria objective, here we do not give priority to the objective achieved by highly ranked jobs, and only care about the maximal tardiness of any job, independent of its hierarchy level.

We show that the problem is optimally solvable. In particular, Algorithm 4, which was shown to be optimal for $1|hierarchy|(T_{\mathcal{J}_1},\dots,T_{\mathcal{J}_c})$, produces a schedule that achieves the minimal tardiness of any job.

Theorem 3.6: Algorithm 4 is optimal also for $1|hierarchy|T_{max}$.

Proof: The proof of Algorithm 3 for $c = 2$, as well as its extension for $c > 2$ (Algorithm 4), are based on exchange arguments. Specifically, every optimal schedule can be modified to a one that agrees with the schedule produced by the algorithm. A close look at the exchange arguments reveals that none of them harms the maximal tardiness of any job in the instance. Thus, if π^* is an optimal schedule with respect to T_{max} it can be converted to agree with the schedule π produced by the algorithm, without harming its feasibility, nor the maximal tardiness. ■

IV. Minimizing Number of Tardy Jobs

In this section we consider the objective function of minimizing the number of tardy jobs. We assume that the number c of different hierarchy levels is a constant. Our results show an interesting distinction between the multi-criteria objective, for which we present an optimal algorithm, and the global objective function, for which we present a hardness proof.

Without a dominance hierarchy, Moore's algorithm is an optimal greedy algorithm for $1||\sum U_j$. A naive approach for the problem with jobs' hierarchy can be based on scheduling the highly rank jobs according to Moore's algorithm, then create spaces between the jobs, such that each job is shifted to complete as close as possible to its due date, and add the jobs of the next level. This approach fails because Moore's algorithm does not take into account the due-dates of the jobs as long as they are not late. The following example demonstrates this issue and highlight the challenges in solving the problem. Let $c = 2$. The dominant set is $\mathcal{A} = \{a_1, a_2\}$, where $p_1 = 2, d_1 = 2$ and $p_2 = 3, d_2 = 4$. The subordinate set consists of a single job $\mathcal{B} = \{b\}$, where $p_b = 1, d_b = 1$. Executing Moore's Algorithm on $\mathcal{A}$ gives the schedule $[a_1, a_2]$. No spacing is possible, thus, b must be late when added. The resulting schedule, π_1, is shown in Figure 5. $U_{\mathcal{A}}(\pi_1) = 1, U_{\mathcal{B}}(\pi_1) = 1$. Note that the same number of tardy jobs is achieved if b is assigned before a_2. The optimal solution for this instance is the schedule π_2, shown in Figure 5. We have $U_{\mathcal{A}}(\pi_2) = 1, U_{\mathcal{B}}(\pi_2) = 0$. Note that the jobs of $\mathcal{A}$ are not processed in EDD order.

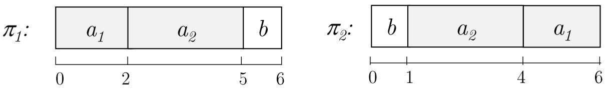


Fig. 5. π_1 : A schedule based on Moore's algorithm. Both a_2 and b are tardy. π_2: An optimal schedule. a_1 is the only tardy job.

A. The multi-criteria objective function: $1|hierarchy|(U_{\mathcal{J}_1},\dots,U_{\mathcal{J}_c})$

In the problem $1|hierarchy|(U_{\mathcal{J}_1},\dots,U_{\mathcal{J}_c})$, the goal is to find a feasible schedule that fulfills the following conditions:

1) The number of tardy jobs from the top hierarchy level, that is, $U_{\mathcal{J}_1}$, is minimal.
2) For every $2 \le \ell \le c$, the number of tardy jobs from the ℓ-th hierarchy level, that is $U_{\mathcal{J}_\ell}$ is the minimal possible among all the schedules that achieve the minimal values of $U_{\mathcal{J}_1},\dots,U_{\mathcal{J}_{\ell-1}}$.

We present an optimal algorithm for a constant number of hierarchy levels. Specifically, we reduce the problem to the problem $1||\sum w_j U_j$, for which an optimal algorithm, based on dynamic programming, is presented in [11].

Algorithm 5 - An optimal algorithm for $1|hierarchy|(U_{\mathcal{J}_1},\dots,U_{\mathcal{J}_c})$, with constant c.

```
Assign every job a weight:
For each $J_i \in \mathcal{J}_c$, let $w_i = 1$.
for $\ell = c-1$ down to 1 do
    $C_\ell = 1 + \sum_{k=\ell+1}^{c} |\mathcal{J}_k| \cdot C_k$
    For each $J_i \in \mathcal{J}_\ell$, let $w_i = 1 + C_\ell$
end for
Ignore the hierarchy levels and find an optimal solution for $1||\sum w_j U_j$ [11].
```

Theorem 4.1: Algorithm 5 is optimal for $1|hierarchy|(U_{\mathcal{J}_1},\dots,U_{\mathcal{J}_c})$ with a constant number of levels.

Proof: The algorithm assign the jobs weight, such that (i) jobs from the same rank have the same weight, and (ii) the weight of each job in hierarchy level ℓ is equal to one plus the total weight of jobs in lower levels. The above weights imply that a single non-tardy job from hierarchy level ℓ contributes to the objective function more than all the jobs in lower levels. Thus, the multi-criteria objective function is achieved by minimizing $\sum w_j U_j$ in the resulting weighted instance.

We show that every optimal solution for $\sum w_j U_j$ corresponds to a feasible solution. Assume by contradiction that a schedule π is optimal for $\sum w_j U_j$, but is not feasible due to job $J_i \in \mathcal{J}_\ell$. This means that J_i can complete on time by delaying jobs from lower ranks. Since the weight of J_i is higher than the total weight of lower rank jobs, we get a contradiction to the π's optimality. ■

The algorithm in [11] assumes that the instance is given as a list of jobs, every job, J_i, is represented by a triplet (p_i, w_i, d_i). In the full version we show how to extend the algorithm to handle a compact representation of the input in which for every weight, w_k we are either given a list of jobs having weight w_k, in which each job is represented by a pair (p_i, d_i); or we are given the amount n_k of jobs having weight w_k, and a single pair (p_k, d_k) such that all n_k jobs having weight w_k have the same processing time p_k, and due-date, d_k.

The fact that an optimal poly-time algorithm exists for instance in the above compact representation gives a nice distinction between the multi-criteria objective and the global objective for which we who a hardness proof already for 4 hierarchy levels.

B. The Global Objective function: $1|hierarchy|\sum U_j$

The goal in the problem $1|hierarchy|\sum U_j$ is to minimize the total number of tardy jobs, independent of their rank. Clearly, the dominance hierarchy plays a significant role also in the global objective problem since it induces the set of feasible schedules.

Theorem 4.2: The problem $1|hierarchy|\sum U_j$ is NP-complete for four or more hierarchy levels.

Proof: Given a schedule, it is possible to calculate the number of non-tardy jobs. Also, Algorithm 2 is a poly-time algorithm for verifying the feasibility of a given schedule, thus, the problem is in NP.

The hardness proof is by a reduction from the subset-sum problem. Given a set of integers $A = \{a_1, a_2, \dots, a_{n_A}\}$ and a target value T, the goal is to decide whether A has a subset $A' \subseteq A$ such that $\sum_{j\in A'} a_j = T$. The subset-sum problem is known to be NP-hard [8].

Given an instance of subset-sum, (A,T), we build an instance of $1|hierarchy|\sum U_j$, consisting of four hierarchy levels, $\mathcal{A},\mathcal{B},\mathcal{C},\mathcal{D}$ as follows.

1) The set $\mathcal{A}$ includes the jobs at the top of the hierarchy. It consists of n_A jobs induced by the subset-sum instance. Specifically, every element $a_j \in A$ contributes to $\mathcal{A}$ one jobs of length a_j and due-date $d_j = T$.
2) The set $\mathcal{B}$ includes a single job, to be denoted J_b, for which $p_b = 2$ and $d_b = T + 1$.
3) The set $\mathcal{C}$ includes T jobs of length 1, all having due-date $d_j = T$.
4) The set $\mathcal{D}$ of lowest level includes T jobs of length $1/T$. All these jobs have due-date $d_j = T + 1$.

We note that the reduction is polynomial assuming a compact representation of the input, that is, the value of T may not be polynomial in n, but we do not list the jobs in $\mathcal{B}$ and $\mathcal{C}$, only specify their amount. Similarly, a schedule can be presented in a compact way - if there are x jobs of length p assigned one after the other, the schedule is presented by specifying x and p, rather than listing all these jobs.

We turn to show the validity of the reduction. The idea is that the jobs of $\mathcal{D}$ whose total length is 1 can be assigned before their due-date only if A is a YES-instance of the subset-sum problem. Intuitively, the job $J_b \in \mathcal{B}$ can benefit from bypassing the jobs of $\mathcal{C} \cup \mathcal{D}$ if and only if there is one more available slot for it in $[0,T]$, and such a slot exists only if the jobs of $\mathcal{A}$ do not use exactly all T slots in $[0,T]$.

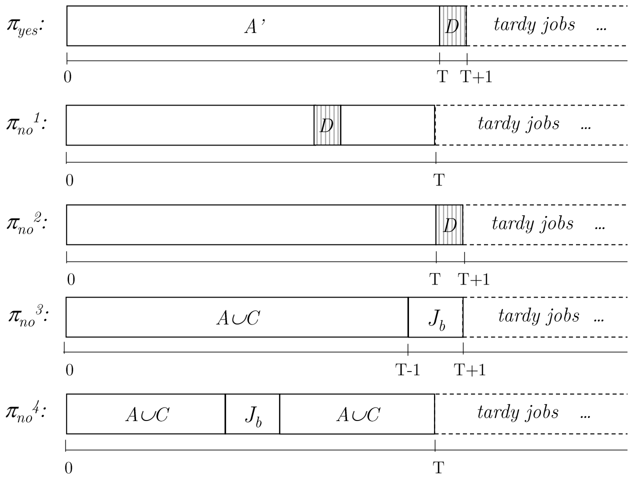


Fig. 6. π_{yes} : An optimal feasible schedule of a YES-instance. π^1_{no} and π^2_{no} : Non-feasible schedules of a NO-instance. π^3_{no} and π^3_{no} : Feasible schedules of a NO-instance.

Claim 4.3: There exists a feasible schedule with more than T non-tardy jobs if and only if A has a subset that sums up to T.

Proof: Assume that A has a subset A' such that $\sum_{j\in A'} a_j = T$. Consider the schedule π_{yes}, depicted in Figure 6, in which the jobs corresponding to the elements of A' are processed in arbitrary order during the interval $[0,T]$ and the T jobs of $\mathcal{D}$ are processed in $[T, T+1]$. All other jobs are late and their schedule is arbitrary. We show that π_{yes} is feasible. The tardy jobs of $\mathcal{A} \cup \mathcal{C}$ all have due-date T. Since the machine processes only jobs from $\mathcal{A}$ in $[0,T]$, and the non-tardy jobs of A' complete their processing exactly at their due-date, it is not possible to add any tardy job to complete on time without harming a non-tardy job from A'. The job J_b, whose due-date is $T+1$ cannot benefit from bypassing the jobs of $\mathcal{D}$, since their total length is 1, and $p_b = 2$. Thus, π_{yes} is a feasible schedule. The number of non-tardy jobs in π_{yes} is $T + |A'|$.

Assume next that A does not have a subset of total sum T. We show that the number of non-tardy jobs in an optimal feasible schedule is at most T. Specifically, we show that the jobs of $\mathcal{D}$ are not processed in any feasible schedule.

Let π_{no} be a schedule of a NO-instance. Consider the interval $[0,T]$. Assume by contradiction that there are more than T non-tardy jobs in π_{no}. Since the jobs of $\mathcal{J} \setminus \mathcal{D}$ all have length at least 1, at most T jobs from $\mathcal{J} \setminus \mathcal{D}$ are processed in $[0,T]$. Also, since $p_b = 2$, if J_b is not tardy, then at most T jobs are processed in $[0,T+1]$. All the jobs that complete after time $T+1$ are clearly tardy. We conclude that if there are more than T non-tardy jobs in π_{no}, then some jobs from $\mathcal{D}$ are non-tardy. Moreover, the jobs of $\mathcal{D}$ are the only jobs whose length is not integral and their total length is 1, therefore, in every optimal feasible solution with some non-tardy jobs from $\mathcal{D}$, all the jobs from $\mathcal{D}$ are non-tardy. Moreover, w.l.o.g., we assume that the jobs of $\mathcal{D}$ are processed sequentially in one time slot in $[0,T+1]$, as otherwise, they can be shifted to be processed sequentially; some other jobs may be shifted to start earlier, which is clearly beneficial for them and therefore does not harm the feasibility of the schedule.

We show that no schedule in which the jobs of $\mathcal{D}$ are allocated one time slot in $[0,T+1]$ is feasible. Assume first that the $\mathcal{D}$-jobs are assigned before time T (see π^1_{no} in Figure 6). If J_b is tardy, then it can remove the jobs of $\mathcal{D}$ and be assigned in $[T-1,T+1]$, resulting in π^3_{no}. If J_b is not tardy, then the jobs of $\mathcal{D}$ will be removed by a tardy job from $\mathcal{C}$, that can assign itself in their slot, resulting in π^4_{no}. This contradicts the feasibility of π^1_{no}.

Assume next that the $\mathcal{D}$-jobs are assigned in $[T,T+1]$ (π^2_{no} in Figure 6). If J_b is tardy, then since a subset of A of total sum T does not exist, at least one job from $\mathcal{C}$ is processed in $[0,T]$. Job J_b can remove this job and the jobs of $\mathcal{D}$ and be assigned in $[T-1,T+1]$, resulting in π^3_{no}. If J_b is not-tardy, then by removing the $\mathcal{D}$-jobs and be processed in $[T-1,T+1]$, J_b can help some tardy job from $\mathcal{C}$ be processed before time T. Again, we get a contradiction to the feasibility of π_{no}.

The above analysis implies that the only possible feasible profiles, have the structure depicted in profiles π^3_{no} or π^4_{no} in Figure 6. The number of non-tardy jobs in these schedules is at most T. ∎

The above claim, together with the fact that subset-sum is NP-hard, implies that $1|hierarchy|\sum U_j$ is NP-hard, already for 4 hierarchy levels. ∎

V. Conclusions

In this paper we analyzed a natural situation in real life scenarios, where some users are more dominant than others, and as a result they should receive a better quality of service. We considered the effect of having such dominance hierarchy on two classical scheduling problems.

We first provided efficient algorithms for testing the feasibility of a schedule, and then considered the problems of (i) minimizing the maximal tardiness of a job, and (ii) minimizing the number of tardy jobs. For the first problem we provided an efficient algorithm for both the bi-criteria objective and the global objective. For the second problem we provided an optimal solution for the bi-criteria objective, and presented a hardness proof for the global objective, when the number of different hierarchy levels in the input set is at least four.

Our work demonstrates the challenges arising in the analysis of systems with users' dominance hierarchy. We believe that this setting represents a natural phenomenon, which be studied further, in additional resource allocation environments.

References

[1] M. Adamu and A. Adewumi. A survey of single machine scheduling to minimize weighted number of tardy jobs. *Journal of Industrial and Management Optimization*, 10:219–241, 2014.

[2] A. Agnetis, F. Rossi, and S. Smriglio. Some results on shop scheduling with s-precedence constraints among job tasks. *Algorithms*, 12(12), 2019.

[3] P. Brucker, A. Gladky, H. Hoogeveen, M. Y. Kovalyov, Chris N. Potts, T. Tautenhahn, and S. L. van de Velde. Scheduling a batching machine. *Journal of Scheduling*, 1(1):31–54, 1998.

[4] I.D. Chase, C. Tovey, and P. Murch, Two's Company, Three's a Crowd: Differences in Dominance Relationships in Isolated versus Socially Embedded Pairs of Fish. *Behaviour.* 140(10):1193–21, 2003.

[5] G. Christodoulou, E. Koutsoupias and A. Nanavati, Coordination mechanisms, *Theor. Comput. Sci.*, 410(36), 2009.

[6] C. Ferguson and K. M. Beaver. Natural born killers: The genetic origins of extreme violence. *Aggression and Violent Behavior.* 14(5):286-–294, 2009.

[7] M. C. Fields and G. N. Frederickson. A faster algorithm for the maximum weighted tardiness problem. *Information Processing Letters*, 36(1):39–44, 1990.

[8] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman, 1979.

[9] L. R. Gesquiere, N. H .Learn, M. C.Simao, P. O. Onyango, S. C. Alberts, and J. Altmann. Life at the top: rank and stress in wild male baboons. *Science (New York, N.Y.)*, 333(6040), 357–360, 2011.

[10] R.L. Graham, E.L. Lawler, J.K. Lenstra, and A.H.G. Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Ann. Discrete Math*, 5:287–326, 1979.

[11] D. Hermelin, S. Karhi, M. Pinedo, and D. Shabtay. New algorithms for minimizing the weighted number of tardy jobs on a single machine. *Annals of Operations Research, Springer*, 298(1):271–287, 2012.

[12] D. Hochbaum and R. Shamir. An $O(nlog^2 n)$ algorithm for the maximum weigthed tardiness problem. *Information Processing Letters*, 31(4):215–219, 1989.

[13] R.M. Karp. Reducibility among combinatorial problems. In *Complexity of computer computations (Proc. Sympos., IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y., 1972)*, pages 85–103, 1972.

[14] E.S. Kim and M. E. Posner. Parallel machine scheduling with s-precedence constraints. *IIE Transactions*, 42(7):525–537, 2010.

[15] E.L. Lawler. Optimal sequencing of a single machine subject to precedence constraints. *Management Sci.*, 19:544–546, 1973.

[16] E.L. Lawler and J.M. Moore. A functional equation and its application to resource allocation and sequencing problems. *Management Sci.*, 16(1):77–84, 1969.

[17] J.M. Moore. An n job, one machine sequencing algorithm for minimizing the number of late jobs. *Management Sci.*, 15:102–109, 1968.

[18] M. Pinedo. *Scheduling: Theory, Algorithms, and Systems.* Springer, 2008.

[19] S. K. Sahni. Algorithms for scheduling independent tasks. *Journal of Assoc. Comput. Mach,*, 23:116–127, 1976.

[20] T. Tamir. Scheduling with bully selfish jobs. *Theory Comput. Syst.*, 50(1):124–146, 2012.

[21] B. Vöcking. *Algorithmic Game Theory*, chapter 20: Selfish Load Balancing. Cambridge University Press, 2007.

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 211–215

DOI: 10.15439/2021F47
ISSN 2300-5963 ACSIS, Vol. 25

A shortened time horizon approach for optimization with differential-algebraic constraints

Paweł Drąg
Department of Control Systems and Mechatronics
Wrocław University of Science and Technology
Wrocław, Poland
pawel.drag@pwr.edu.pl

***Abstract*—In this work a new numerical optimization scheme based on a shortened time horizon approach was designed. The shortened time horizon strategy has never been presented or tested numerically. The new methodology was applied for a single objective optimization task subject to a system of nonlinear differential-algebraic constraints, which can take a form of differential-algebraic equations (DAEs). Moreover, it was assumed, that an application of a cooperated multiple shooting with direct solution method, like direct shooting approach, does not enable us to solve the DAE system, even on relatively small subintervals. Therefore, the new solution procedure is based on two main parts: i) designing of an alternative differential-algebraic constraints, ii) parametrization of a new constraints system by the multiple shooting approach and further simulation of the alternative system independently on small subintervals. Then, the simulation interval can be modified by the shortened time horizon approach. The presented algorithm was used to solve a highly nonlinear optimization task of a fed-batch reactor operation.**

***Index Terms*—numerical optimization, differential-algebraic constraints, shortened time horizon, multiple shooting method**

I. Introduction

The appropriate treatment of the nonlinear constraints can enable us to implement new numerical procedures, helpful in the model-based simulations [3], [4], [9]. In this work, the attention is paid on an optimization task with differential-algebraic constraints, which can take a form of differential-algebraic equations (DAEs). Classically, the systems of the nonlinear DAE constraints can be solved with the multiple shooting approach [6], [7], [11]. Unfortunately, even the multiple shooting methods can fail, when initial conditions are far from the solution trajectory [1], [2], [5]. Therefore, a shortened time horizon (STH) approach was considered as a tool to influence a difficulty of a nonlinear optimization task. The combination of the multiple shooing method with the STH approach can be treated as a base to design a new efficient optimization method subject to the nonlinear differential-algebraic constraints.

This article is constructed as follows. In Section 2 the shortened time horizon approach for DAE constraints is introduced. Then, in Section 3, the new solution procedure is presented. The results of numerical computations are discussed in Section 4. Finally, the presented considerations are summarized in Section 5.

II. The shortened time horizon for differential-algebraic constraints

In this work, a system of the nonlinear differential-algebraic constraints is considered

$$(DAE) \qquad \left\{ \begin{array}{rcl} \dot{y}(t) & = & f(y(t), z(t), u(t), p, t) \\ 0 & = & g(y(t), z(t), u(t), p, t) \\ t & \in & [t_0 \quad t_f] \\ y(t_0) & = & y_0 \\ z(t_0) & = & z_0 \end{array} \right. \tag{1}$$

where the state of the DAE constraints (1) is represented by a vector of differential variables $y(t) \in \mathcal{R}^{n_y}$ and a vector of algebraic variables $z(t) \in \mathcal{R}^{n_z}$ with $\dot{y}(t) = \frac{dy(t)}{dt}$. Moreover, $u(t) \in \mathcal{R}^{n_u}$ denotes a vector of input functions. A vector of model parameters constant in time is represented by $p \in \mathcal{R}^{n_p}$, $t \in \mathcal{R}$ is an independent variable and a range of t is known *a priori*. The functions f and g are of C^2 class and

$$\begin{array}{ll} f: & \mathcal{R}^{n_y} \times \mathcal{R}^{n_z} \times \mathcal{R}^{n_u} \times \mathcal{R}^{n_p} \times \mathcal{R} \rightarrow \mathcal{R}^{n_y} \\ g: & \mathcal{R}^{n_y} \times \mathcal{R}^{n_z} \times \mathcal{R}^{n_u} \times \mathcal{R}^{n_p} \times \mathcal{R} \rightarrow \mathcal{R}^{n_z} \end{array} \tag{2}$$

The shortened horizon approach is based on an appropriate modification of a considered range of the assumed independent variable. The name of this method reflects, that in many cases *time* is considered as the natural independent variable. The shortened time approach is motivated, that the number of shooting intervals does not have to be known *a priori*. Therefore, the range of the independent variable is shortened according to the computational algorithms capabilities. Then, with a known solution obtained for a shortened range, the calculation can be continued iteratively for wider ranges of the independent variable.

Assumption 1. The range of the independent variable $t \in [t_0 \quad t_f]$ can be modified and parametrized by $q \in [0 \quad 1]$ in the shortened horizon approach as $t \in [q \cdot t_0 \quad q \cdot t_f]$.

This research was funded by the Department of Control Systems and Mechatronics at Wrocław University of Science and Technology.

Therefore, the formulation of the considered constraints (1) in the context of the shortened time horizon method takes a new form

$$\Big(DAE(q)\Big) \quad \begin{cases} \dot{y}(t) &= f(y(t), z(t), u(t), p, t) \\ 0 &= g(y(t), z(t), u(t), p, t) \\ t &\in [qt_0 \quad qt_f] \\ y(t_0) &= y_0 \\ z(t_0) &= z_0 \end{cases} \tag{3}$$

A direct shooting approach is one of a common used method to simulate the systems described by the nonlinear DAE constraints on the assumed interval of the independent variable (3). The mentioned approach is based on a multiple shooting, where a range of the independent variable $t \in [t_0 \quad t_f]$ is divided on an assumed number N subintervals

$$t^i \in [qt_0^i \quad qt_f^i], \qquad i = 1, \dots, N \tag{4}$$

where

$$qt_0 = qt_0^1 < qt_f^1 = qt_0^2 < \dots < qt_f^{N-1} = qt_0^N < qt_f^N. \tag{5}$$

Then, the DAE constraints (3) can be considered on each subinterval independently such, that

$$\Big(DAE^i(q)\Big) \begin{cases} \dot{y}^i(t^i) &= f^i(y^i(t^i), z^i(t^i), u^i(t^i), p, t^i) \\ 0 &= g^i(y^i(t^i), z^i(t^i), u^i(t^i), p, t^i) \\ t^i &\in [qt_0^i \quad qt_f^i] \\ y(t_0^i) &= y_0^i \\ z(t_0^i) &= z_0^i \end{cases} \tag{6}$$

for $i = 1, \dots, N$. The variables $y^i(t), z^i(t), u^i(t)$ and p have a similar interpretation like in eq. (1). Moreover, the multiple shooting approach enable us a parametrization of both the state variables $y^i(t^i)$ and $z^i(t^i)$, as well as the input function $u^i(t^i)$, for $i = 1, \dots, N$.

Assumption 2. On the given subinterval of the independent variable $t^i \in [t_0^i \quad t_f^i]$, the trajectory of the differential state variable $y^i(t^i)$ can be parametrized and represented by a new state variable $\widetilde{y}^i(t^i)$ modeled by a system of linear differential equation of the form

$$\dot{\widetilde{y}}^i(t^i) = D^i \widetilde{y}^i(t^i), \tag{7}$$

for $i = 1, \dots, N$, where D^i is a $n_{y^i} \times n_{y^i}$ diagonal matrix.

Assumption 3. On the given subinterval of the independent variable $t^i \in [t_0^i \quad t_f^i]$, the trajectory of the algebraic state variable $z^i(t^i)$ can be parametrized and represented by a new state variable $\widetilde{z}^i(t^i)$ modeled by a system of linear algebraic equation of the form

$$0 = \widetilde{z}^i(t^i) - (A_{z^i} t^i + b_{z^i}), \tag{8}$$

for $i = 1, \dots, N$, where A_{z^i} is a $n_{z^i} \times n_{z^i}$ diagonal matrix and b_{z^i} is a vector with n_{z^i} elements.

Assumption 4. On the given subinterval of the independent variable $t^i \in [t_0^i \quad t_f^i]$, the trajectory of the input function $u^i(t^i)$ can be parametrized and represented by a new input function $\widetilde{u}^i(t^i)$ modeled by a piecewise constant function of the form

$$0 = \widetilde{u}^i(t^i) - (b_{u^i}), \tag{9}$$

for $i = 1, \dots, N$, where b_{u^i} is a vector with n_{u^i} elements.

Unfortunately, in general, the values of the initial conditions for state variables in shooting points are unknown

$$\mathbf{x_{y_0 z_0}} = \begin{bmatrix} y_0^1 & z_0^1 \\ \vdots & \vdots \\ y_0^N & z_0^N \end{bmatrix}. \tag{10}$$

Therefore, the unknown initial conditions $\mathbf{x_{y_0 z_0}}$ can be treated as an important part of decision variables in a nonlinear optimization task. Moreover, the parametrization variables, which define the trajectories of the state variables and the input function, can be used to built a matrix of decision variables

$$\mathbf{X} = \begin{bmatrix} y_0^1 & z_0^1 & d(A_{z^1}) & b_{z^1} & b_{u^1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_0^N & z_0^N & d(A_{z^N}) & b_{z^N} & b_{u^N} \end{bmatrix}, \tag{11}$$

where an operator $d(B)$ denotes the diagonal elements of the matrix B.

To solve the new system of continuous differential-algebraic constraints, it is enough to assign arbitrary values to the decision variables matrix (11)

$$\Big(\widetilde{DAE}^i(q)\Big) \quad \begin{cases} \dot{\widetilde{y}}^i(t^i) &= D^i \widetilde{y}^i(t^i) \\ 0 &= \widetilde{z}^i(t^i) - (A_{z^i} t^i + b_{z^i}) \\ 0 &= \widetilde{u}^i(t^i) - (b_{u^i}) \\ t^i &\in [qt_0^i \quad qt_f^i] \\ \widetilde{y}^i(t_0^i) &= y_0^i \\ \widetilde{z}^i(t_0^i) &= A_{z^i} t_0^i + b_{z^i} \\ \widetilde{u}^i(t_0^i) &= b_{u^i} \end{cases} \tag{12}$$

for $i = 1, \dots, N$.

To determine the matrix of decision variables (11), the additional pointwise algebraic constraints were imposed. They were used to force a continuity of the obtained state trajectories $\widetilde{y}(t)$ and $\widetilde{z}(t)$, ensure consistent initial conditions, as well as provide such models of original DAE constraints, that the dynamics of the obtained solutions will meet the primary constraints (1).

This approach results with a system of pointwise equality constraints consisted with concatenated vectors of specified restriction types

$$G(\mathbf{X}) = \begin{bmatrix} G_{cont}(\mathbf{X}) \\ G_{cons}(\mathbf{X}) \\ G_{dyn}(\mathbf{X}) \end{bmatrix} = 0, \tag{13}$$

where

$$G_{cont}(\mathbf{X}) = \begin{bmatrix} \widetilde{y}^1(t_f^1) - y_0^2 \\ \vdots \\ \widetilde{y}^{N-1}(t_f^{N-1}) - y_0^N \end{bmatrix}, \tag{14}$$

is a vector of the continuity constraints,

$$G_{cons}(\mathbf{X}) = \begin{bmatrix} g^1(y_0^1, \widetilde{z}^1(t_0^1), \widetilde{u}^1(t_0^1), p, t_0^1) \\ \vdots \\ g^N(y_0^N, \widetilde{z}^N(t_0^N), \widetilde{u}^N(t_0^N), p, t_0^N) \end{bmatrix}, \tag{15}$$

is a vector of the consistency constraints, as well as

$$G_{dyn}(\mathbf{X}) =$$

$$= \begin{bmatrix} D^1 y_0^1 - f^1(y_0^1, \widetilde{z}^1(t_0^1), \widetilde{u}^1(t_0^1), p, t_0^1) \\ \vdots \\ D^N y_0^N - f^N(y_0^N, \widetilde{z}^N(t_0^N), \widetilde{u}^N(t_0^N), p, t_0^N) \end{bmatrix}, \tag{16}$$

is a vector of the dynamical constraints.

Corollary 5. The $\widetilde{DAE^i}(q)$ model (12) is a special case of a linear differential-algebraic system with time-dependent coefficients

$$\begin{bmatrix} \dot{\widetilde{y}}(t) \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} D & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} \widetilde{y}(t) \\ \widetilde{z}(t) \\ \widetilde{u}(t) \end{bmatrix} - \begin{bmatrix} 0 \\ A_z t + b_z \\ b_u \end{bmatrix}. \tag{17}$$

The proposed methodology can be used to transform the system of highly nonlinear differential-algebraic constraints (1) into the system of linear differential-algebraic constraints (12) with pointwise algebraic constraints related with the continuity, consistency and dynamical constraints (13). The new system of constraints can be considered in the context of nonlinear optimization task

$$\min_{\mathbf{X}} F(\mathbf{X}), \tag{18}$$

where F is a scalar-valued objective function. The additional assumptions related to the F function need to be reflected in a chosen numerical optimization procedure used further in an inner loop of the shortened time horizon optimization algorithm.

III. The new algorithm

The steps of the new optimization approach were presented as the Shortened Time Horizon Optimization (STHO) Algorithm. The designed Algorithm requires some input information, which should be supplied in the Preliminary steps. The defined sequence $\{q_k\}_{k=0}^n$ determines a progress of the shortened time approach (step P 1 in the STHO Algorithm). In the other words, the interval of the independent variable in the kth outer loop iteration is $[q_k t_0 \quad q_k t_f]$. Moreover, the first value $q_0 = 0$ is defined and no more used. Then, the assumed number of shooting subintervals N is constant during the performed computations (P 2). To start the algorithm, the considered DAE constraints (1) are necessary to be inserted (P 3). Moreover, the values of n_y, n_z and n_u with a range of the independent variable $t \in [t_0 \quad t_f]$ are used to obtain the parametrized model of the differential-algebraic constraints. To start the outer loop, the matrix of the initial solution (11) needs to be defined (P 4).

The STHO Algorithm

```
Preliminaries
P 1   Define a sequence of $n+1$ elements $\{q_k\}_{k=0}^n$,
      where $q_0 = 0$ and $q_n = 1$.
P 2   Define a number of shooting subintervals $N \in \mathbb{N}^+$.
P 3   Define a system of $DAE$ constraints (1) with
      values $n_y$, $n_z$, $n_u$ and a range of the independent
      variable $t \in [t_0 \quad t_f]$.
P 4   Choose the initial solution matrix $\mathbf{X_0}$ (11).

The outer loop
FOR $k = 1, 2, \ldots, n$
O 1     Choose a value $q_k$.
O 2     Define subintervals $t^i \in [q_k t_0^i \quad q_k t_f^i]$,
        where $i = 1, \ldots, N$.
O 3     Define the new constraints models $\widetilde{DAE^i}(q_k)$.
O 4     Define the vector of algebraic constraints $G(\mathbf{X})$.

        The inner loop
          Find $\mathbf{X}^\star$ by solving the optimization task

              $\min_{\mathbf{X}} F(\mathbf{X})$
          subject to
              $\widetilde{DAE^i}(q_k)$,    $i = 1, \ldots N$,
              $G(\mathbf{X}) = 0$

O 5     The obtained solution $\mathbf{X}^\star$ is the new initial
        solution for the next iteration of the outer loop
        $\mathbf{X}_0 = \mathbf{X}^\star$
END-FOR.
```

The optimization method is consisted of two main parts, which will be referred to as an outer- and inner loop. In the inner loop, a numerical optimization procedure solves a parametrized task subject to the differential-algebraic constraints (12), as well as additional equality restrictions (13) resulting from the multiple shooting method. The shortened time approach is a base for the outer loop of the new algorithm. It can take a form of a „for" iterations, where a q parameter is incremented according to the assumed way. The solution obtained as a result at a current iteration of the outer loop, is a starting point for the inner loop in the next outer iteration.

The outer loop is mainly concentrated around a nonlinear optimization task constructing for a given value q_k (O 1). In the steps (O 2) and (O 3), the appropriate subintervals t^i with the new models $\widetilde{DAE^i}$ are defined. Then, the DAE constraints and models $\widetilde{DAE^i}$ (P 3) will be used to calculate the system of pointwise algebraic constraints $G(\mathbf{X})$ (O 4). The constraints $G(\mathbf{X})$ represent similarity between the designed model (12) and the original DAE constraints (1).

The current task is solved in the inner loop by a chosen numerical optimization procedure. The solver can cooperate with a numerical integrator of the differential-algebraic constraints, although the new system of constraints (12) can be solved analytically in many cases. A selection of efficient numerical algorithms for constrained optimization is presented in [10].

IV. Computational experiments

The algorithm designed in this study was implemented in Matlab environment and applied for solving optimization task of searching for optimal operation of a fed-batch reactor. The considered model is consisted on the differential and algebraic state variables

$$y(t) = [y_1(t) \quad y_2(t) \quad y_3(t) \quad y_4(t) \quad y_5(t)]^T \tag{19}$$

$$z(t) = [z_1(t) \quad z_2(t) \quad z_3(t)]^T, \tag{20}$$

as well as the objective function

$$\max_{u(t)} \quad y_1(t_f)y_5(t_f) \tag{21}$$

The differential-algebraic constraints with initial conditions for the differential state trajectories

$$y(t_0) = [0.0 \quad 0.0 \quad 1.0 \quad 5.0 \quad 1.0]^T \tag{22}$$

are based on the work of Luus and Rosen [8]. The initial conditions for the algebraic state variables $z(t_0)$ can be calculated based on the initial conditions of $y(t_0)$ (22). Moreover, the input function $u(t)$ is constrained by a pair of inequalities

$$0.0 \leq u(t) \leq 10.0 \tag{23}$$

The final value of the state variable y_4 is bounded by

$$y_4(t_f) \leq 14.35 \tag{24}$$

The process duration range was assumed and equal $t \in [0 \quad 25]$. The presented objective function (21) subject to the continuous differential-algebraic constraints, as well as the pointwise constraint (24), was parametrized by a direct shooting method with $N = 25$ subintervals. The parametrization resulted with new decision variables and continuity constraints.

The optimization task with the appropriate parametrization and introduced constraints was solved in three different ways

- case 1: minimization of the objective function extended by a penalty function,
- case 2: optimization with interior-point algorithm implemented in *fmincon* Matlab function,
- case 3: solution by the STHO algorithm presented in this work.

A. The task parametrization

According to the classical multiple shooting rules, the initial conditions of the differential state trajectories are treated as new decision variables. Then, the input function $u(t)$ was parametrized as a piecewise constant trajectory. Therefore, the vector of decision variables $\mathbf{X}$ was consisted of 146 elements. Moreover, to ensure the continuity of the obtained solution, additional 121 equality constraints were take into account. In this set of the decision variables and constraint functions, one decision variable together with one continuity constraint was introduced to represent the inequality (24). This is a basic multiple shooting parametrization used in numerical experiments in the case 1 and case 2.

In the case 3, the parametrization appropriate for the designed STHO algorithm was applied. At the beginning, the initial conditions of the differential state trajectories $\widetilde{y}(t)$, as well as the input function $\widetilde{u}(t)$ are parametrized in the same way like in the cases 1 and 2. Then, the additional variables were introduced to obtain parametrized systems $\widetilde{DAE^i}$: $n_y \times N = 5 \times 25 = 125$ variables for matrices D^i and $n_z \times 2 \times N = 3 \times 2 \times 25 = 150$ variables to parametrize the algebraic state trajectory $\widetilde{z}(t)$.

Moreover, 121 equality continuity constraints $G_{cont}(\mathbf{X})$, $3 \times 25 = 75$ equality consistency constraints $G_{cons}(\mathbf{X})$, as well as $5 \times 25 = 125$ equality dynamical constraints $G_{dyn}(\mathbf{X})$ were introduced. Finally, the optimization task with 421 decision variables, as well as the equality and box constraints was considered.

B. Numerical results

The simulations were started with a similar approach, to the one presented in the article [8]. In the case 1, the objective function was in a form of minimized penalty function

$$\min_{\mathbf{X}} \mathcal{J}_1 = -y_1^{25}(t_f)y_5^{25}(t_f) + \rho\|G_{cont}(\mathbf{X})\|_2^2, \tag{25}$$

where $\|\cdot\|_2$ denotes a l_2 norm and ρ is a penalty parameter. In performed calculation $\rho = 10^4$. The value of the obtained objective function was equal to $\mathcal{J}_1(\mathbf{X}^\star) = -112.71$.

In the case 2, the considered task was taken a form

$$\min_{\mathbf{X}} \mathcal{J}_2 = -y_1^{25}(t_f)y_5^{25}(t_f) \tag{26}$$

subject to

$$G_{cont}(\mathbf{X}) = 0. \tag{27}$$

The solution vector $\mathbf{X}^\star$ was obtained by the interior-point algorithm implemented in the Matlab's *fmincon* function. The obtained value of the minimized objective function was equal to $\mathcal{J}_2(\mathbf{X}^\star) = -112.6231$.

The computational calculations performed in the case 3 were more time-consuming and indicated on some benefits, as well as disadvantages of the STHO algorithm. First of all, the algorithm was working in the outer loop implemented as

for q_k from 0.1 to 1.0 with a step 0.1

The main problem, meet at the beginning of the computations at the first iteration of the outer loop, was to indicate the initial solution near to a such local minimizer, which can fulfill all the constraints $G(\mathbf{X}) = 0$ based on the $\widetilde{DAE^i}(q_k)$ solution. For the $\mathbf{X}_0$ near the local minimizer, the solution was obtained and extended in the next iterations of the outer loop. The figs. 1-2 show the state trajectories $\widetilde{y}_1(t)$ and $\widetilde{y}_2(t)$ obtained for the initial solution near to the local minimizer and calculated for different values of q_k.

The main drawback of the presented solution is related to the construction of $\widetilde{DAE^i}(q_k)$. The models of the linear differential-algebraic constraints systems with variable coefficients result with solutions of a form $Ae^{\lambda t}$ for the differential state variables. Therefore, small modification in the vector of decision variables resulted in significant changes in the solution trajectories. Therefore, the new model of the differential-algebraic constraints $\widetilde{DAE^i}(q_k)$ can show comparable computational difficulties, like an original one (1).

The obtained solution trajectories seems to be piecewise linear, especially, if larger number of subintervals is considered. Therefore, the value $\lambda \approx 0$. This is particular true, if the solution is calculated for higher values of the independent variable.

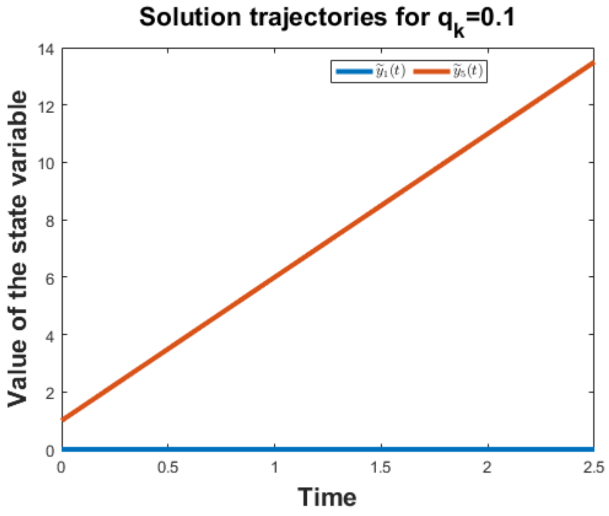


Fig. 1. The trajectories of the state variables $\widetilde{y}_1(t)$ and $\widetilde{y}_5(t)$ obtained for $q_k = 0.1$.

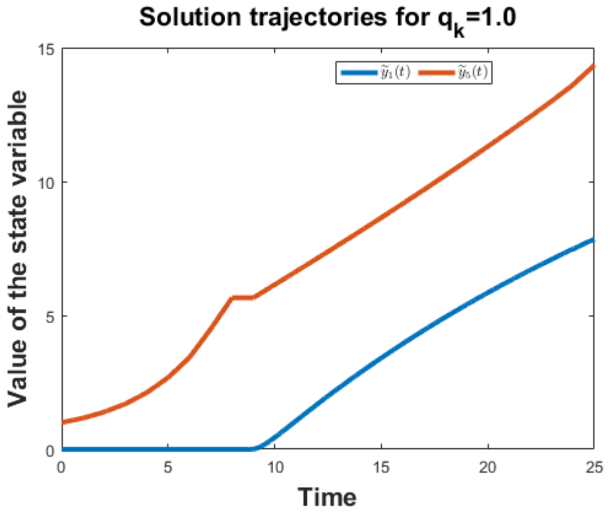


Fig. 2. The trajectories of the state variables $\widetilde{y}_1(t)$ and $\widetilde{y}_5(t)$ obtained for $q_k = 1.0$.

V. Conclusion

In the presented work the STHO Algorithm for optimization with the differential-algebraic constraints was presented. The designed shortened time horizon approach was based on the multiple shooting method, as well as implemented in two main parts - outer in inner iterations. The outer iteration generates the assumed number of subintervals and new constraints models with appropriate vector of pointwise algebraic constraints. In the inner loop, the defined nonlinear optimization tasks with modeled $\widetilde{DAE^i}(q_k)$ constraints and additional equality constraints is solved by a chosen numerical optimization procedure. The final solution of the inner loop is further treated as an initial solution for the next iteration in the outer loop.

The designed algorithm was used to solve the optimization task, where an optimal operation of the fed-batch reactor should be found. The performed computations indicated benefits and drawbacks of the designed procedure. The solution trajectories can be found and simply extended on the wider subintervals, if the appropriate initial solution is known *a priori*.

Acknowledgment

The author would like to thank the anonymous reviewers for thoroughly reading the manuscript and providing valuable comments.

References

[1] J.T. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, SIAM, Philadelphia, 2010, https://doi.org/10.1137/1.9780898718577.

[2] B. Beykal, M. Onel, O. Onel, E.N. Pistikopoulos, *A data-driven optimization algorithm for differential algebraic equations with numerical infeasibilities*, AIChE Journal, vol. 66, 2020, article no. e16657, https://doi.org/10.1002/aic.16657.

[3] B. Burnak, E.N. Pistikopoulos, *Integrated process design, scheduling, and model predictive control of batch processes with closed-loop implementation*, AIChE Journal, vol. 66, 2020, article no. e16981, https://doi.org/10.1002/aic.16981.

[4] A. Caspari, L. Lüken, P. Schäfer, Y. Vaupel, A. Mhamdi, L.T. Biegler, A. Mitsos, *Dynamic optimization with complementarity constraints: Smoothing for direct shooting*, Computers and Chemical Engineering, vol. 139, 2020, article no. 106891, https://doi.org/10.1016/j.compchemeng.2020.106891.

[5] P. Drąg, *A Direct Optimization Algorithm for Problems with Differential-Algebraic Constraints: Application to Heat and Mass Transfer*, Applied Sciences, vol. 10, 2020, art. no. 9027, pp. 1-19, https://doi.org/10.3390/app10249027.

[6] P. Drąg, K. Styczeń, *The new approach for dynamic optimization with variability constraints*. In: S. Fidanova (ed.), Recent advances in computational optimization : results of the Workshop on Computational Optimization WCO 2017. Cham, Springer, 2019. pp. 35-46, https://doi.org/10.1007/978-3-319-99648-6_3.

[7] M.T. Kelley, R. Baldick, M. Baldea, *A direct transcription-based multiple shooting formulation for dynamic optimization*, Computers and Chemical Engineering, vol. 140, 2020, art. no. 106846, https://doi.org/10.1016/j.compchemeng.2020.106846.

[8] R. Luus, O. Rosen, *Application of dynamic programming to final state constrained optimal control problems*, Industrial & Engineering Chemistry Research, vol. 30, 1991, pp. 1525-1530.

[9] D. Pandelidis, M. Drąg, P. Drąg, W. Worek, S. Cetin, *Comparative analysis between traditional and M-Cycle based cooling tower*, International Journal of Heat and Mass Transfer, vol. 159, 2020, art. no. 120124, pp. 1-13, https://doi.org/10.1016/j.ijheatmasstransfer.2020.120124.

[10] J. Nocedal, S. Wright, *Numerical Optimization*. Springer, New York, NY, 2006, https://doi.org/10.1007/978-0-387-40065-5.

[11] D.M. Yancy-Caballero, L.T. Biegler, R. Guirardello, *Large-scale DAE-constrained optimization applied to a modified spouted bed reactor for ethylene production from methane*, Computers and Chemical Engineering, vol. 113, 2018, pp 162-183, https://doi.org/10.1016/j.compchemeng.2018.03.017.

DOI: 10.15439/2021F74
ISSN 2300-5963 ACSIS, Vol. 25

Solving assignment problems via Quantum Computing: a case-study in train seating arrangement

Ilaria Gioda*, Davide Caputo†, Edoardo Fadda*, Daniele Manerba‡,
Blanca Silva Fernández†, and Roberto Tadei*

*Department of Control and Computer Engineering, Politecnico di Torino, 10129 Torino, Italy
Email: ilaria.gioda@studenti.polito.it, {edoardo.fadda, roberto.tadei}@polito.it

†Data Reply s.r.l., 10126 Torino, Italy
Email: {da.caputo, b.silvafernandez}@reply.it

‡Department of Information Engineering, Università degli Studi di Brescia, 25123 Brescia, Italy
Email: daniele.manerba@unibs.it

***Abstract*—In recent years, researchers have oriented their studies towards new technologies based on quantum physics that should resolve complex problems currently considered to be intractable. This new research area is called Quantum Computing. What makes Quantum Computing so attractive is the particular way with which quantum technology operates and the great potential it can offer to solve real-world problems. This work focuses on solving assignment-like combinatorial optimization problems by exploiting this novel computational approach. A case-study, denoted as the Seating Arrangement Optimization problem, is considered. It is modeled through the Quadratic Unconstrained Binary Optimization paradigm and solved through two tools made available by the *D-Wave Systems* company, QBSolv, and a quantum-classical hybrid system. The obtained experimental results are compared in terms of solution quality and computational efficiency.**

I. Introduction

COMBINATORIAL Optimization (CO) is one of the most studied research fields in the area of optimization. The application of this research area extends to many sectors, and more and more researchers are active in model and solve effectively and efficiently the problems belonging to this category. Among others, one of the most recent and innovative modeling approaches to formulate a CO problem is the so-called Quadratic Unconstrained Binary Optimization (QUBO) paradigm. Among the various approaches for solving combinatorial optimization problems in the QUBO form, in recent years, researchers have begun to be oriented towards a new computational frontier, as the *Quantum Computing*. This paper focuses on analyzing this new computational approach, specifically for the resolution of assignment problems. We analyze the Seating Arrangement Optimization problem (SAOP) as a case-study, which was first formulated as a QUBO problem and then solved through the use of some tools made available by *D-Wave Systems*, a Canadian company specializing in quantum computing. In particular, quantum systems suitable for solving optimization problems are called quantum annealers; they exploit the physical concept that everything in nature tends to evolve towards equilibrium (see [1]). It is worth noting that there exist alternative ways to implement quantum-like algorithms as the one explored in [4].

The remaining part of this paper is organized as follows. Section II describes the Quadratic Unconstrained Binary Optimization (QUBO) paradigm and reports on the existing solvers dedicated to problems in this particular form, including the quantum technologies offered by *D-Wave Systems*. Section III presents the case-study considered, the Seating Arrangement Optimization problem. The problem is first described and modeled as a quadratic problem. Then an equivalent QUBO formulation is derived. Section IV describes and compares the computational results of the experimental analysis. Section V provides conclusions and a brief discussion on possible future works.

II. Quantum Computing solvers

The leader company that works with quantum annealers is *D-Wave Systems Inc.*. In particular, this organization deals with building and studying quantum technologies and, for some years, has allowed external people to use their quantum annealers to solve specific commercial problems, especially combinatorial optimization ones. Quantum annealers are designed to solve complex combinatorial optimization problems in a particular formulation, the Quadratic Unconstrained Binary Optimization (QUBO) one. The goal of a QUBO model is to find an optimal solution by minimizing an objective function in the form

$$\min_{\mathbf{x}\in\{0,1\}^{|N|}} \mathbf{x}^T Q \mathbf{x} \tag{1}$$

where $\mathbf{x}$ is a column vector of binary variables of size $|N|$ and Q an upper-triangular $|N| \times |N|$ matrix, called QUBO matrix. Not all optimization problems come in this form. However, many of them can be rewritten as a QUBO model. The constraints identified for the problem must be readjusted and converted into penalties to form the actual objective function (1) that has to be minimized. Specifically, as in classical

Lagrangean relaxations, the purpose of these penalties is to prevent the optimizer from choosing solutions that violate the constraints. They involve the addition of a positive quantity, therefore not favorable to the minimization objective in case of infeasible solutions [3]. Some standard ways of creating this translation from classical constraints can be found in [3]. By using this formulation, two solution methods are available: *QBSolv* and *D-Wave Systems*.

QBSolv is an open-source solver released in January 2017, which runs on the CPU like traditional solvers. Its goal is to solve significant QUBO problems with high connectivity. The solver strategy consists of partitioning significantly large QUBO problems into smaller components and applying a specified sampling method (the classical Tabu Search algorithm, by default) independently to each of these pieces to find the minimum value required for the optimization. Further technical details on QBSolv can be found in [2].

D-Wave Systems allows to submit and solve a problem modeled as QUBO on a remote quantum computer. To do this, in 2018, the computing company made available to users a cloud service, the *D-Wave's Leap*, and a set of Python APIs, the Solver API (SAPI), that allow any developer to access and submit any problem to the *D-Wave Quantum System*.

III. A case-study: The Seating Arrangement Optimization Problem

The considered case-study focuses on passenger transport on high-speed trains considering the Italian Government's new regulations on social distancing due to the COVID-19 pandemic.

The railway companies have currently adapted their passenger positioning strategies by embracing a seating arrangement as a "checkerboard" pattern, i.e., with the allocation of passengers to alternate seats, to counter the spread of the COVID-19 virus. Nevertheless, with the adoption of this strategy, the filling capacity of the wagons has dropped to 50% of the total capacity, leading to a drastic reduction in the high-speed rail operators' earnings. This is due to the mismatch between the costs necessary for activating the railway transport lines and the revenues obtained from ticket sales. From now on, we will denote the examined case-study as the Seating Arrangement Optimization problem.

The objective of the Seating Arrangement Optimization problem (SAOP) is to fill the train wagon as much as possible within the restrictions on social distancing due to the COVID-19 health emergency. Still, it aims to maximize the number of passengers belonging to the same family or living group in adjacent seats. Although the focus of the problem can be extended to the entire train, the study refers to only one wagon. Then, for a multi-wagon train, the procedure will be run for each wagon separately. Furthermore, we assumed a static situation: just one train segment, i.e., a trip between two adjacent stations, is considered so that the number of passengers and their social relationships are known beforehand without any changes during the travel.

Some fundamental elements characterize the SAOP. A set of passengers that has to be transported on a high-speed train is given. During the ticket reservation procedure, each passenger is associated with a unique identifier, the booking ID, which can be shared or not with other passengers. The important assumption of the problem is that people with the same booking ID belong to the same family or living group. This condition, therefore, assumes they can be excluded from the social distancing impositions prescribed by the regulations against the spread of the COVID-19 virus. A high-speed train's wagon is then considered. The wagon has a certain number of seats. Each seat is represented by a pair of coordinates, a row and a column number, which collocate it into a grid. Finally, it is necessary to consider the following requirements:

- allocation of one and only one seat to each one of the considered passengers (avoid that a passenger has more than one seat assigned to him);
- allocation of one passenger at most to each seat (avoid different passengers being assigned the same seat);
- allocation of not adjacent (in front/behind/left/right) seats to people belonging to different families (identified by different booking IDs).

Let us consider the following sets and parameters:

- $R = \{1, 2, \dots, r_{max}\}$: set of seats row numbers;
- $C = \{1, 2, \dots, c_{max}\}$: set of seats column numbers;
- K: set of booking IDs;
- n_k: nb. of passengers with the same booking ID $k \in K$.

Moreover, let us define the variable

$$x_{(r,c),k} := \begin{cases} 1 & \text{if a passenger with booking ID } k \text{ is} \\ & \text{assigned to seat with row and column } (r,c) \\ 0 & \text{otherwise} \end{cases}$$

for each row $r \in R$, column $c \in C$, and booking ID $k \in K$. Then, a natural quadratic programming model for the SAOP can be stated as:

$$\max \sum_{k} \sum_{(r,c)} x_{(r,c),k} \cdot x_{(r+1,c),k} + \\ + \sum_{k} \sum_{(r,c)} x_{(r,c),k} \cdot x_{(r,c+1),k} \quad (2)$$

subject to

$$\sum_{(r,c)} x_{(r,c),k} = n_k, \quad k \in K \quad (3)$$

$$\sum_{k} x_{(r,c),k} \leq 1, \quad r \in R, \ c \in C \quad (4)$$

$$x_{(r,c),k} \cdot x_{(r+1,c),k'} = 0, \\ r \in R \setminus \{r_{max}\}, \ c \in C, \ k, k' \in K, k \neq k' \quad (5)$$

$$x_{(r,c),k} \cdot x_{(r,c+1),k'} = 0, \\ r \in R, \ c \in C \setminus \{c_{max}\}, \ k, k' \in K, k \neq k' \quad (6)$$

$$x_{(r,c),k} \in \{0, 1\}, \quad r \in R, \ c \in C, \ k \in K. \quad (7)$$

The objective function (2) maximizes the number of passengers with the same booking ID assigned to adjacent seats. Constraints (3) state that each passenger with a given booking ID is assigned to one seat, while constraints (4) state that each seat is assigned to at most one passenger with a given booking ID. Constraints (5) ensure that two seats, one next to the other (in the same column), are not assigned to passengers with different booking IDs, while constraints (6) ensure that two seats, one in front of the other (in the same row), are not assigned to passengers with different booking IDs. Finally, binary conditions on the variables are stated in (7).

A. QUBO formulation

Since the QUBO paradigm asks for an unconstrained model, as the one in (1), the constraints (3)-(6) and the cost function (2) are relaxed and aggregated into a single objective function through non-negative parameters λ's, to be calibrated (see later). In particular, we chose to set these parametric coefficients as numerical and to associate each of them with a specific group of constraints presented in model (2)–(7). We decided to adopt this modeling choice to minimize the number of λ parameters needed, as they represent a non-negligible obstacle during the model calibration.

To do this relaxation, we built a penalty term for each of the identified constraints by following the approach from [3]. Hence, a QUBO formulation for the SAOP problem becomes:

$$\min \lambda_A H_A + \lambda_B H_B + \lambda_C H_C + \lambda_D H_D - H_E \tag{8}$$

where

- the penalty term associated with constraints (3) is
$$H_A = \sum_k (n_k - \sum_{(r,c)} x_{(r,c),k})^2$$
- the penalty term associated with constraints (4) is
$$H_B = \sum_{(r,c)} \sum_{k,k'} x_{(r,c),k} \cdot x_{(r,c),k'}$$
- the penalty term associated with constraints (5) is
$$H_C = \sum_{(r,c)} \sum_{k,k'} x_{(r,c),k} \cdot x_{(r+1,c),k'}$$
- the penalty term associated with constraints (6) is
$$H_D = \sum_{(r,c)} \sum_{k,k'} x_{(r,c),k} \cdot x_{(r,c+1),k'}$$
- the penalty term associated with objective function (2) is
$$H_E = \sum_k \sum_{(r,c)} x_{(r,c),k} \cdot x_{(r+1,c),k} + \sum_k \sum_{(r,c)} x_{(r,c),k} \cdot x_{(r,c+1),k}.$$

Note that, unlike the other penalties, a squaring has been introduced in H_A as it is necessary to be able to grasp the relationship between the values assumed by different variables within the solution.

Starting from (8), the Q matrix of model (1) has been derived. To do that, we need to identify the relationship between the problem's variables. First of all, the single QUBO terms of the function (8) need to be expanded. Then, after the coefficients have been found, they are multiplied by the parametric coefficients λ_A, λ_B, λ_C and λ_D, whose purpose is to give more or less weight to each QUBO penalty such that the constraints are imposed when searching for the solution.

IV. Computational results

This section reports the results obtained by executing several instances of the SAOP modeled as a QUBO using the two tools offered by *D-Wave Systems*, namely, the QBSolv and D-Wave Leap's cloud-based quantum-classical hybrid solver (from now on referred to as D-Wave Hybrid Solver). Initially, the problem size in terms of the number of variables is reported. Then, the two solvers are compared in terms of optimal solutions and computational times.

An ad-hoc data set containing simulated test instances about seats, passengers, and bookings were created for the performed experiments. The input that we provided to our QUBO model has been created based on an indicative estimate of realistic data of a high-speed train. In particular, it was decided to use a wagon consisting of 80 seats, placed in a 4×20 grid, made up of 4 horizontal (the rows) and 20 vertical (the columns) rows. Taking as a reference a reasonable number of passengers for a high-speed train, 1000 passengers have been created. Still, only a small subset of them was used for our restricted experimental analysis. In particular, for the experiments reported in the following, the maximum number of people that have been tested is 52. Since it was necessary to associate a specific booking ID to each passenger, we decided to use 300 distinct booking IDs to make a reasonably homogeneous assignment. The final range of assigned booking IDs is 290 booking IDs, and the minimum number of passengers with the same booking ID is 1 while the maximum is 8. All the experiments have been carried out on a desktop computer with a 1.8 GHz Intel Core i7-8550U processor.

A. Quality of solutions

The quality of the *D-Wave Systems* solvers is now analyzed. After having calibrated the λ parameters in model (8), by using the Python APIs of the Ocean SDK, the two solvers were used to solve different instances of the analyzed problem. The numerical results for the SAOP various instances can be seen in Table I. For each problem instance (identified by "Seats", "Passengers", "Distinct booking IDs" columns), we report in the "Total minimum energy" column the value of the best solution found by each solver (i.e., the minimum value of the expression (8)). Moreover, the number of passengers allocated to seats inside the train wagon (fifth column) and the number of people with the same booking ID correctly assigned to adjacent seats (sixth column) are reported for each solution.

The two solvers seem to perform well, most of the time reaching the goal of allocating people with the same booking

TABLE I
OPTIMAL SOLUTIONS OBTAINED BY RUNNING THE QUBO MODEL INSTANCES WITH THE *D-Wave Systems* SOLVERS

Seats	Passengers	Distinct booking IDs	Solver	Nb. of passengers with an assigned seat	Nb. of passengers with same booking ID assigned to adjacent seats	Total minimum energy
80	11	3	QBSolv	11	10	-464.300
			D-Wave Hybrid	10	11	-464.300
80	16	4	QBSolv	16	15	-682.800
			D-Wave Hybrid	16	15	-682.800
80	19	5	QBSolv	19	18	-762.000
			D-Wave Hybrid	19	18	-762.000
80	23	7	QBSolv	23	21	-849.400
			D-Wave Hybrid	23	21	-849.400
80	28	8	QBSolv	28	26	-1067.900
			D-Wave Hybrid	28	26	-1067.900
80	34	9	QBSolv	34	32	-1382.000
			D-Wave Hybrid	34	32	-1382.000
80	39	11	QBSolv	39	37	-1496.700
			D-Wave Hybrid	39	37	-1496.700
80	44	13	QBSolv	44	41	-1646.900
			D-Wave Hybrid	44	41	-1646.900
80	50	14	QBSolv	50	45	-1947.500
			D-Wave Hybrid	50	47	-1958.300
80	51	15	QBSolv	51	46	-1953.000
			D-Wave Hybrid	51	47	-1961.100

ID to adjacent seats. Furthermore, an improvement compared to the passenger transport's current situation has been achieved. Both solvers manage to find at least an acceptable seating arrangement up to 15 booking IDs for a total of 51 passengers, bringing therefore to have a filling percentage of the seats up to 63,75% (instead of the classical 50%).

For most instances, the D-Wave Hybrid Solver finds solutions with the same energy as those found by QBSolv. This means that the solver running on the CPU performs well in solution quality, even without quantum hardware usage. However, there are two cases, i.e., the ones corresponding to the instances with 14 and 15 distinct booking IDs (respectively 50 and 51 passengers), where D-Wave Hybrid Solver finds two lower energy and better solutions than those found by QBSolv. The computational time of QBSolv ranges from 1.2s (for the instance with 11 passengers and 3 distinct booking IDs) up to 18.8s (for the instance with 51 passengers and 15 different booking IDs). Instead, if the D-Wave Hybrid Solver is used for solving the same problems, the computational time ranges from 6.5s to 22.1s. The difference between them lies in how they work: QBSolv works locally on the CPU while D-Wave Hybrid Solver requires remote access via the Internet to a physically remote system shared between multiple users.

V. CONCLUSIONS

This paper has analyzed how assignment-like combinatorial optimization problems can be effectively solved through quantum technology tools. Specifically, we aimed to investigate this innovative computation technique, quantum computing, and explore the advantages and disadvantages that derive from it.

We considered a specific case-study concerning the allocation of passengers to seats on high-speed trains with the recent hygiene and health regulations on social distancing due to the COVID-19 pandemic. The experiments show that the quantum approach is a feasible way to solve the problem effectively.

ACKNOWLEDGMENTS

This study has been proposed by and developed with the contribution of the Quantum Computing Team of Data Reply S.r.l., Torino (Italy).

REFERENCES

[1] Marchenkova A., *"What's the difference between quantum annealing and universal gate quantum computers?"*, https://medium.com/quantum-bits/what-s-the-difference-between-quantum-annealing-and-universal-gate-quantum-computers-c5e5099175a1

[2] Booth, M. & Reinhardt, S.P. *"Partitioning Optimization Problems for Hybrid Classical / Quantum Execution"*, Technical Report (2017)

[3] Glover, F., Kochenberger, G. & Du, Y. *"Quantum Bridge Analytics I: a tutorial on formulating and using QUBO models."* 4OR-Q J Oper Res 17, 335–371 (2019). https://doi.org/10.1007/s10288-019-00424-y

[4] S.B. Hengeveld, N. Rubiano da Silva, D.S. Gonçalves, P.H. Souto Ribeiro, A. Mucherino, Solving the One-dimensional Distance Geometry Problem by Optical Computing, arXiv e-print, arXiv:2105.12118, version 1, May 2021.

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 221–225

DOI: 10.15439/2021F66
ISSN 2300-5963 ACSIS, Vol. 25

Worst-Case Analysis of an Approximation Algorithm for Single Machine Scheduling Problem

Natalia Grigoreva
St.Petersburg State University
Universitetskay nab. 7/9, St.Petersburg, Russia
Email: n.s.grig@gmail.com

***Abstract*—The problem of minimizing the maximum delivery times while scheduling jobs on the single processor is a classical combinatorial optimization problem. This problem is denoted by $1|r_j, q_j|C_{\max}$, has many applications, and it is NP-hard in strong sense. The goal of this paper is to propose a new 3/2-approximation algorithm, which runs in $O(n \log n)$ time. We proved that the bound of 3/2 is tight. To check the efficiency of the algorithm we tested it on random generated problems of up to 5000 jobs.**

Keywords: single-machine scheduling problem, release and delivery times, approximation algorithm, worst-case performance ratio

I. Introduction

WE CONSIDER a set of jobs $U = \{1, 2, \dots, n\}$. Each job i must be processed without interruption for $t_i > 0$ time units on the processor, which can process at most one job at time. Each job i has a release time $r_i \geq 0$, when the job is ready for processing, and a delivery time $q_i \geq 0$. The delivery of each job begins immediately after processing has been completed.The objective is to minimize the time, by which all jobs are delivered. In the notation of Graham *et al.*[5] this problem is denoted by $1|r_j, q_j|C_{\max}$, and has many applications.

It is required to construct a schedule, that is, to find for each job $i \in U$ the start time τ_i, provided that $r_i \leq \tau_i$. The goal is to construct a schedule that minimizes $C_{\max} = \max\{\tau_i + t_i + q_i | i \in U\}$, which is the delivery time of the last job.

In [11] it is shown that the problem is NP-hard in the strong sense, but there are exact polynomial algorithms for some special cases. Some authors considered an equivalent formulation of the problem, in which instead of the delivery time for each job, the due date $D_i = K - q_i$, is known, where K is a constant, and the objective function is the maximum lateness $L_{\max} = \max\{\tau_i + t_i - D_i | i \in U\}$. This formulation of the problem is denoted as $1|r_i|L_{\max}$. The advantage of the model with delivery times is that the value of the objective function is always positive, while the maximum lateness can be negative or equal to zero.

If we swap the delivery times and the release times, we get an inverse problem with the property that the solution of the direct problem $S = (i_1, i_2, \dots, i_n)$ is optimal if and only if the permutation $S_{inv} = (i_n, i_{n-1}, \dots, i_1)$ is the optimal solution of the inverse problem.

The $1|r_j, q_j|C_{\max}$ is the main subproblem in many important models of scheduling theory, such as flowshop and job-shop problems, multiprocessor scheduling and online single-machine scheduling [12].The study of this problem is of theoretical interest and is useful in practical industrial application [1], [4], [18].

Several approximation algorithms are known for solving the problem $1|r_i, q_i|C_{\max}$.

The first algorithm for constructing an approximation schedule is the Schrage heuristic [17] - an extended Jackson rule, which is formulated as follows: each time the processor is free, a ready job with the maximum delivery time is assigned to it. Computational complexity of the Schrage heuristic is $O(n \log n)$, the algorithm is a 2-approximation algorithm [10].

K. Potts [16] proposed an algorithm in which the extended Jackson's rule algorithm repeats n times. Computational complexity of the Potts algorithm is $O(n^2 \log n)$. The worst-case performance ratio is equal 3/2.

L. Hall and D. Schmois [8] have developed the method in which the Potts algorithm is applied to the direct and inverse problem. In total, the algorithm builds $4n$ schedules and chooses the best one. Computational complexity of the algorithm is $O(n^2 \log n)$. The worst-case performance ratio is equal 4/3.

E. Novitsky and K. Smutnitsky [14] proposed an 3/2-approximation algorithm, which creates only two permutations. For the first time, the Jackson rule is applied, then the interference job is determined and the set of jobs is divided into two sets: jobs that should be performed before the interference job in the order of their release times, and after it that should be performed after it in non-increasing delivery times. The best schedule is selected from two schedules. Computational complexity is $O(n \log n)$.

All the mentioned algorithms use the list greedy Schrage algorithm as a basic heuristic.

The works of [2], [3], [13], [15] developed branch and bound algorithms for single processor scheduling problem using different branching rules and bounding techniques. The first efficient algorithm is Carlier algorithm [3], which optimally solves instances with up to thousand of jobs. This algorithm constructs a full solution by extended Jackson's rule in each node of the search tree.

One way to improve the performance of the branch and bound method is to use approximation efficient algorithms to obtain upper bounds. Such algorithms should have a good approximation ratio and the low computational complexity.

IEEE Catalog Number: CFP2185N-ART ©2021, PTI

One of the popular scheduling tools are list algorithms that build non-delayed schedules. In the list algorithm, at each step, the job with the highest priority is selected from the set of ready jobs. But the optimal schedule may not belong to the class of non-delayed schedules.

IIT schedules (IIT - inserted idle time) were defined in [9] as feasible schedules in which the processor can be idle when there are jobs ready to run.

The author considered the IIT 2-approximation algorithm [6] and developed the branch and bound algorithm for single-machine scheduling problem with receive and delivery times [7]. The main idea of greedy algorithms for solving this problem is the choice at each step of the highest priority job, before the execution of which the processor could be idle.

In this paper, we propose an approximation algorithm ICA for solving the problem, which creates two permutations, one by the Schrage method, and the second by an algorithm with inserted idle time. The construction of each permutation requires $O(n \log n)$ action.

The article is organized as follows: Section 2 presents a new approximate algorithm scheduling IJR and the combined ICA algorithm, which builds two permutations and chooses the best one. We prove that the worst-case performance ratio of the ICA algorithm is equal 3/2 and this bound is tight in section 3. The results of the computational experiment, which showed the speed and practical accuracy of the algorithm, are given in Section 4. In conclusion, the main results obtained in the article are formulated.

II. IJR and ICA scheduling algorithms.

First, we describe the IJR scheduling algorithm.

The main idea of the IJR algorithm is that sometimes it is better to place a priority job on service, even if it leads to some idle time of the processor.

In the IJR algorithm two jobs are selected: the highest priority job and the highest priority from ready jobs. The paper has established special conditions in which it is advantageous to organize the unforced idle time of the processor. These conditions allow to choose between two jobs.

The algorithm IJR is a greedy algorithm, but not a list algorithm and can be used as a basic heuristic for various scheduling models and constructing a branch and bound method.

We introduce the following notation: $S_k = (i_1, i_2, \dots, i_k)$ is the partial schedule, $time = \max\{\tau_i + t_i | i \in S_{k-1}\}$ is the time to release the processor after the execution of already scheduled jobs. We store ready jobs in the queue with priorities Q_1, the priority of a job is it's delivery time.

A. Algorithm IJR

1) Sort all jobs in non-descending order of release times: $r_{j_1} \le r_{j_2} \le \dots \le r_{j_n}$.
2) Define the lower bound of the objective function
$LB_1 = \min\{r_i | \ i \in U\} + \sum_{i=1}^{n} t_i + \min\{q_i \mid i \in U\}$.
$LB_2 = \max\{r_i + t_i + q_i \mid i \in U\}$.
$LB = \max\{LB_1, LB_2\}$.

Algorithm 1 IJR algorithm (the main loop)

```
1:  Initialize: $S_0 = \emptyset$; $Q_1 = \emptyset$; $l \leftarrow 1$;
2:  for $k \leftarrow 1$ to $n$ do
3:      if $Q_1 = \emptyset$; then
4:          $time \leftarrow r_{j_l}$;
5:      end if
6:      while $r_{j_l} \le time$ do
7:          Add ready job $j_l$ to the $Q_1$ queue; $l \leftarrow l + 1$;
8:      end while
9:      Select the ready job $u \in Q_1$ with the maximum delivery
        time $q_u = \max\{q_i | i \in Q_1\}$;
10:     $r_{up} \leftarrow time + t_u$;
11:     while $r_{j_l} < r_{up}$ do
12:         if $(q_{j_l} \ge LB/2) \& (q_{j_l} - q_u \le r_{j_l} - time)$ then
13:             Set job $j_l$ on the processor: $S_k \leftarrow S_{k-1} \cup \{j_l\}$;
                $\tau(j_l) \leftarrow r(j_l)$; $time \leftarrow \tau(j_l) + t(j_l)$;
                $l \leftarrow l + 1$; break;
14:         else
15:             $j_l$ is added to the queue $Q_1$; $l \leftarrow l + 1$;
16:         end if
17:     end while
18:     Set job $u$ on the processor: $S_k \leftarrow S_{k-1} \cup \{u\}$;
        $\tau(u) \leftarrow time$; $time \leftarrow \tau(u) + t(u)$;
        delete $u$ from $Q_1$;
19: end for
20: The schedule $S_n$ and its makespan is equal $C_{\max}(S_n)$.
```

B. Combined scheduling algorithm ICA

1. Construct the schedule S_{JR} by the Schrage algorithm, denote the makespan of the schedule $C_{\max}(S_{JR})$.

2. Construct the schedule S by the IJR algorithm, denote the makespan of the schedule $C_{\max}(S)$.

3. Choose the schedule S_A with a smaller value of the objective function: $C_{\max}(S_A) = \min\{C_{\max}(S), C_{\max}(S_{JR})\}$.

Computational complexity of the algorithm is $O(n \log n)$. The algorithm ICA constructs two permutations: one by the Schrage algorithm, the computational complexity of which is $O(n \log n)$, and one by the IJR algorithm.

Let's show that for the IJR algorithm the computational complexity is equal $O(n \log n)$. First, we sorts all jobs in non-descending order of it's release times, this step requires $O(n \log n)$ actions.

The main operation is to select a job from a set of ready jobs. We store the ready jobs as a priority queue Q_1, which can be organized as a binary heap, the priority of job j is the delivery time q_j. At the steps 6-8 of the algorithm, we add new ready jobs to the queue Q_1 such that $r_{j_i} \le time$, adding each job requires $O(\log n)$ actions. The job u with the highest priority is selected for $O(1)$ actions (step 9).

At steps 11-17 we add new jobs j_i to the queue Q_1, for which $r_{j_i} < time + t_u$. If there is a job j_l for which all conditions (step 12) are met, then we map j_l on the processor and go to the beginning of the main cycle (step 2). Otherwise, we look through all candidates by placing them in the queue

Q_1, and map the job u on the processor on the step 18. Then the job u is deleted from queue Q_1, which requires $O(\log n)$ actions.

Each job can be added to the queue at most once: building the binary heap requires $O(n \log n)$ actions. The total computational complexity is equal $O(n \log n)$.

III. Properties of the schedule constructed by the algorithm ICA

The properties of the schedule created by the combined algorithm ICA proposed in Section 2 are formulated and proved in the following lemmas.

Let the IJR algorithm constructs a schedule S, the makespan is equal to $C_{\max}(S)$, and the schedule S_{JR} is constructed by the JR algorithm, the makespan is equal to $C_{\max}(S_{JR})$. Consider some definitions that were introduced in [16] for schedules constructed according to Jackson's rule, and which are important characteristics for IIT schedules.

Definition 3.1: [16] A critical job is a job j_c such that $C_{\max}(S) = \tau_{j_c} + t_{j_c} + q_{j_c}$. If there are several such jobs, then we choose the earliest one in the schedule S.

Definition 3.2: [16] A critical sequence in a schedule S is a sequence of jobs $J(S) = (j_a, j_{a+1}, \ldots, j_c)$ such that j_c is the critical job and there is no processor idle time in the schedule, starting from the start of the job j_a until the job j_c ends.

The job j_a is either the first job in the schedule, or the processor is idle before it.

Definition 3.3: [16] A job j_u in a critical sequence is called interference job if $q_{j_u} < q_{j_c}$ and $q_{j_i} \geq q_{j_c}$, for $i > u$.

Proposition 3.4: [16] If for all jobs of the critical sequence it is true that $r_{j_i} \geq r_{j_a}$ and $q_{j_i} \geq q_{j_c}$, then the schedule is optimal.

Let us introduce a definition of delayed job that can be encountered in IIT schedules.

Definition 3.5: A job j_v from a critical sequence $J(S) = (j_a, j_{a+1}, \ldots, j_c)$ is called a delayed job if $r_{j_v} < r_{j_a}$.

An interference job can be a delayed job.

Let us formulate two properties of the IJR schedule, similar to the properties of JR schedules [16].

Lemma 3.6: If there is the interference job j_u in a critical sequence, then $C_{\max}(S) - C_{\max}(S_{opt}) < t_{j_u}$

Lemma 3.7: If there is no any delayed jobs in the critical sequence, then $C_{\max}(S) - C_{\max}(S_{opt}) \leq q_{j_c}$.

The proof of the lemmas is similar to [16], in Lemma 3.7 it is necessary to add the condition that there are no delayed jobs.

Let us introduce the following notation: if J is some sequence of jobs, then $T(J) = \sum_{i \in J} t_i$ and $r_{\min}(J) = \min\{r_i | i \in J\}$.

Lemma 3.8: If the interference job j_u in the critical sequence $J(S) = (S_1, j_u, S_2)$ is executed after the sequence S_2 in an optimal schedule or between j_a and j_c, then $C_{\max}(S)/C_{\max}(S_{opt}) \leq 3/2$.

Proof: If $t_{j_u} \leq C_{\max}(S_{opt})/2$, then the lemma 3.8 is true by lemma 3.6.

Let $t_{j_u} > C_{\max}(S_{opt})/2$. If the interference job j_u is executed after all jobs of the sequence S_2 in an optimal schedule, then $C_{\max}(S_{opt}) \geq r_{\min}(S_2) + T(S_2) + t_{j_u} + q_{j_u}$.

Then

$$C_{\max}(S) - C_{\max}(S_{opt}) \leq r_{j_a} + T(S_1) + t_{j_u} + T(S_2) +$$

$$+ q_{j_c} - r_{min}(S_2) - T(S_2) - t_{j_u} - q_{j_u} =$$

$$= r_{j_a} + T(S_1) - r_{min}(S_2) + q_{j_c} - q_{j_u} =$$

$$= -idle + q_{j_c} - q_{j_u}.$$

Where $idle = -r_{j_a} - T(S_1) + r_{min}(S_2) > 0$. If $q_{j_c} < LB/2$, then the lemma has proven. Let $q_{j_c} \geq LB/2$. Choose a job $v \in S_2$ such that $r_v = r_{\min}(S_2)$. Then $q_v \geq LB/2$, and $idle = r_v - time > q_v - q_{j_u} \geq q_{j_c} - q_{j_u}$.

Then $C_{\max}(S) - C_{\max}(S_{opt}) < LB/2$.

If in the optimal schedule, the job j_u is performed between jobs j_a and j_c, then

$$C_{\max}(S_{opt}) \geq r_{j_a} + t_{j_u} + T(S_2) + q_{j_c}.$$

Therefore $C_{\max}(S) - C_{\max}(S_{opt}) \leq r_{j_a} + T(S_1) + t_{j_u} + T(S_2) + q_{j_c} - r_{j_a} - t_{j_u} - T(S_2) - q_{j_c} = T(S_1) < LB/2$. ■

Lemma 3.9: Let the schedule S_{JR} is constructed by the JR algorithm. There is the interference job j_u in the critical sequence $J(S_{JR}) = (F_1, j_u, F_2)$.

If the interference job j_u is executed before all jobs of the sequence F_2 in an optimal schedule, then $C_{\max}(S_{JR})/C_{\max}(S_{opt}) \leq 3/2$.

Proof: If $t_{j_u} \leq C_{\max}(S_{opt})/2$, then the lemma 3.9 is true by lemma 3.6. If the job j_u is executed before all jobs of the sequence F_2 in an optimal schedule S_{opt}, then

$$C_{\max}(S_{opt}) \geq r_{z_a} + t_{j_u} + T(F_2) + q_{z_c}.$$

Then $C_{\max}(S_{JR}) - C_{\max}(S_{opt}) \leq T(F_1) < LB/2$. ■

Theorem 3.10: The algorithm ICA constructs a schedule S_A for which $C_{\max}(S_A)/C_{\max}(S_{opt}) \leq 3/2$.

Proof: Let the schedule S be constructed using the IJR algorithm and the schedule S_{JR} be constructed by the JR algorithm. There are the critical sequence $J(S) = (j_a, j_{a+1}, \ldots, j_c)$ in S, and the critical sequence $J(S_{JR}) = (z_a, z_{a+1}, \ldots, z_c)$ in S_{JR}.

We consider all the possible cases.

Case 1. There are no interference and delayed jobs in $J(S)$ or no interference job in $J(S_{JR})$ critical sequences.

In this case, the corresponding algorithm has constructed an optimal schedule.

Case 2. There are interference jobs in each critical sequence. It is required to consider the case in which two interference jobs is the same large job such that $t_{j_u} > C_{\max}(S_{opt})/2$.

The makespan of the schedule S_{JR} is equal to $C_{\max}(S_{JR}) = r_{z_a} + T(J(S_{JR})) + q_{z_c}$.

If delivery time of the critical job z_c does not exceed $C_{\max}(S_{opt})/2$, then by Lemma 3.7 the theorem is true. Let $q_{z_c} > C_{\max}(S_{opt})/2$.

By virtue of the proved Lemmas 3.8 and 3.9, it suffices to consider the case in which in the optimal schedule, the job j_u should be carried out after all jobs of the sequence F_2 and before the job j_a.

TABLE I
RELEASE, PROCESSING AND DELIVERY TIMES OF JOBS FROM U

Job	r_i	t_i	q_i
x	ε	ε	$M-2*\varepsilon$
a	$M/2-\varepsilon$	ε	$M/2$
u	0	$M/2+\varepsilon$	0
c	$M/2+\varepsilon$	ε	$M/2-2*\varepsilon$

According to the properties of the IJR algorithm, the processor is idle until the time r_{j_a} and $q_{j_a} > LB/2$.

Then $C_{\max}(S_{opt}) \geq r_{\min}(F_2) + T(F_2) + t_{j_u} + t_{j_a} + q_{j_a}$. Hence,

$$C_{\max}(S_{JR}) - C_{\max}(S_{opt}) \leq r_{z_a} + T(J(S_{JR})) + q_{z_c} -$$

$$-r_{\min}(F_2) - T(F_2) - t_{j_u} - t_{j_a} - q_{j_a} =$$

$$= r_{z_a} + T(F_1) + q_{z_c} - r_{\min}(F_2) - t_{j_a} - q_{j_a} <$$

$$< q_{z_c} - q_{j_a} < LB/2.$$

This is true because $q_{z_c} < LB$ and $q_{j_a} > LB/2$. In this case, the Schrage algorithm constructs a 3/2 approximation schedule.

Case 3. There is the interference job in the critical sequence $J(S_{JR})$ and there are some delayed jobs in $J(S)$.

If there is no an interference job in the critical sequence $J(S)$, then $q_i \geq q_{j_c}$ for all jobs from the critical sequence $i \in J(S)$. But in the critical sequence there are jobs that can be started before the job j_a.

Then $C_{\max}(S_{opt}) \geq r(J(S)) + T(J(S)) + q_{j_c}$. Hence

$$C_{\max}(S) - C_{\max}(S_{opt}) \leq r_{j_a} + T(J(S)) + q_{j_c} - r(J(S)) -$$

$$-T(J(S)) - q_{j_c} = r_{j_a} - r(J(S)) < LB/2.$$

We have proven that the worst-case performance ratio of ICA algorithm is equal 3/2. ■

Lemma 3.11: There is an example for which the ratio $C_{\max}(S_A)/C_{\max}(S_{opt})$ tends to 3/2.

Proof: Consider a system of four jobs $U = \{x, a, u, c\}$. The data for the system of jobs are given in Table 1, where M is a constant. The lower bound for the objective function is $LB = M$.

The IJR algorithm constructs the schedule $S = (x, a, u, c)$. The processor is idle $M/2 - \varepsilon$ time units before starting the job a. The objective function is equal $C_{\max}(S) = 3/2M$. The JR algorithm constructs the schedule $S_{JR} = (u, x, a, c)$. The objective function is equal $C_{\max}(S_{JR}) = 3/2M - \varepsilon$.

The optimal schedule is $S_{opt} = (x, u, a, c)$, the value of the objective function is equal $C_{\max}(S_{opt}) = M + \varepsilon$. When ε tends to zero, the ratio $C_{\max}(S_A)/C_{\max}(S_{opt})$ tends to 3/2. ■

IV. COMPUTATIONAL EXPERIMENT

To find out the practical efficiency of the algorithm, a computational experiment was carried out. The goals of the computational experiment were comparison of the accuracy of the IJR algorithm and of the JR Schrage algorithm and comparison of the accuracy of the combined ICA algorithm with the accuracy of the NS algorithm of Novitsky and Smutnitsky using random test examples.

The initial data was generated by the method described by Carlier [3], the same method generating of test examples were used by Novitsky and Smutnitsky when they compared their proposed algorithms with Hall and Schmois and Schrage algorithms. For each job i, three integer values were chosen with uniform distribution : q_i and r_i between 1 and nK. There were chosen the values for K from 10 to 25, which were noted by Carlier as the most difficult for the problem under consideration. For each value of n and K, we considered 100 instances. Three groups of examples were considered. The processing times for each of the groups were selected from the following intervals ($t_{\max} = 50$):

1) Type A: t_j from $[1, t_{\max}]$,
2) Type B: t_j from $[1, t_{\max}/2]$, for $j \in 1 : n-1$ and t_n from $[nt_{\max}/8, 3nt_{\max}/8]$,
3) Type C: t_j from $[1, t_{\max}/3]$, for $j \in 1 : n-2$ and t_{n-1}, t_n from $[nt_{\max}/12, 3nt_{\max}/12]$.

Groups of type B contains instances with one long job and groups of type C contains instances with two long jobs.

The value of the objective function $C_{\max}$ was compared with the optimal value of the objective function C_{opt}, which was obtained by the branch and bound method [7]. In all tables, n is the number of jobs in the instance.

For tests of type A, n were changed from 50 to 5000 and for all tests the value $K = 20$ was chosen. For tests of type A, the IJR algorithm generates more optimal solutions than the NS and JR algorithms. The JR algorithm very rarely receives optimal solution. The average relative error of the solution is small for all algorithms and decreases with increasing n.The average relative error is 0.03 percent for the IJR algorithm, 0.97 percent for the JR algorithm and 0.2 percent for the NS algorithm (for $n = 50$). For $n = 5000$ the average relative error is 0.001 percent for the IJR algorithm, 0.01 percent for the JR algorithm and 0.002 percent for the NS algorithm.

The results of experiments in which we change the constant K, from 10 to 22 does not significantly affect the results of the algorithms for instances of Type A.

The theoretical analysis of the algorithms shows that the most difficult examples take place when there are one or two long jobs. Such tests were generated in groups of type B and type C.

Tables 2 and 3 show the results of comparison of algorithms for tests of type B. For these groups of tests, we considered the combined ICA, in which the best solution was chosen of the two solutions, obtained by the JR and IJR algorithms.

TABLE II
TYPE B. THE NUMBER OF OPTIMAL SOLUTIONS.

n	K	N_{IJR}	N_{JR}	N_{NS}	N_{ICA}
100	10	51	23	25	58
100	14	29	47	48	62
100	15	25	48	49	59
100	16	53	21	34	69
100	18	46	46	49	71
100	20	24	15	16	33
100	22	44	29	33	57

TABLE III
TYPE B. THE AVERAGE RELATIVE ERROR OF ALGORITHMS.

n	K	R_{IJR}	R_{JR}	R_{NS}	R_{ICA}
100	10	1.02	1.05	1.04	1.005
100	14	1.06	1.03	1.03	1.004
100	15	1.05	1.04	1.04	1.007
100	16	1.01	1.04	1.01	1.004
100	18	1.05	1.04	1.03	1.005
100	20	1.05	1.06	1.03	1.007
100	22	1.02	1.03	1.03	1.006

Columns 3—6 of Table 2 show the number of tests (in percent) for which optimal solutions were generated by algorithms IJR, JR, NS and ICA, respectively.

Table 2 show that for tests of type B the number of optimal solutions for the ICA algorithm is greater then the number of optimal solutions for the NS algorithm.

The value of the average relative error of algorithms for tests of type B are given in the Table 3. Columns 3-6 of Table 3 show the value of the average relative error $R_A = C_{\max}(S_A)/C_{opt}$ of algorithms IJR, JR , NS and ICA, respectively.

The relative error of the solution increases for all algorithms JR, IJR, NS and it is from 1 to 6 percent on average. The author's ICA algorithm has significantly more advantages. It combines the advantages of the Schrage algorithm, which does not allow unforced idle time and IJR algorithm, which allows them. The relative error of the solution for ICA algorithm is from 1.004 to 1.007 on average, but the relative error of NS algorithm is from 1.01 to 1.04.

The worst solutions for the JR algorithm had a relative error of 23 percent, for the IJR algorithm - 19 percent but for the combined ICA algorithm had only 7 percent. No test was received during testing, for which both JR and IJR algorithms have constructed a solution with large relative error. The combined algorithm generates two permutations just like the NS algorithm, but its average relative error is significantly less, and the number of optimal solutions obtained is greater.

V. CONCLUSION

The paper considers the problem of scheduling for single processor with release and delivery times. The paper proposes a new 3/2 approximation algorithm with computational complexity $O(n \log n)$, in which the priority of the job is taken into account first and processor can be idle, when certain conditions are met. The example is given, which shows that the bound of 3/2 is tight. The computational experiment has confirmed the practical efficiency of the algorithm.

REFERENCES

[1] C. Artigues and D.Feillet, "A branch and bound method for the job-shop problem with sequence-dependent setup times," *Ann. of Oper. Res.,* vol. 159, 2008, pp. 135–159, http://dx.doi.org/10,1287/opre.49.6.854.10014.

[2] K.R. Baker, Introduction to Sequencing and Scheduling. John Wiley & Son, New York, 1974.

[3] J. Carlier, "The one machine sequencing problem," *European Journal of Operational Research*, vol.11, 1982, pp.42—47, http://dx.doi.org/10.1016/s0377-2217(82)80007-6.

[4] C. Chandra, Z. Liu, J. He, T. Ruohonen, "A binary branch and bound algorithm to minimize maximum scheduling cost," *Omega* , vol. 42, 2014, pp. 9–15, http://dx.doi.org/10.1016/j.omega2013.02.005.

[5] R.L. Graham, E.L. Lawler, J.K. Lenstra and A.H.G. Rinnooy Kan, "Optimization and approximation in deterministic sequencing and scheduling: A survey," *Ann. of Disc. Math.* ,vol. 5, no. 10, 1979, pp. 287–326, http://dx.doi.org/10.1016/S0167-5060(08)70356-X.

[6] N. Grigoreva, "Single Machine Inserted Idle Time Scheduling with Release Times and Due Dates," *Proc. DOOR2016. Vladivostoc. Russia. Sep.19-23.2016.* Ceur-WS, vol. 1623, 2016, pp. 336–343.

[7] N.S Grigoreva, "Single Machine Scheduling with Precedence Constrains, Release and Delivery times", *in Proceedings of 40th Anniversary International Conference on Information Systems Architecture and Technology–ISAT 2019- Part III.* (Advances in Intelligent Systems and Computing, v. 1052), pp. 188 –198, http://dx.doi.org/10.1007/978-3-030-30443-0-17.

[8] L.A. Hall and D.B. Shmoys, "Jackson's rule for single-machine scheduling: making a good heuristic better," *Mathematics of Operations Research*, 17 (1) , 1992, pp. 22–35.

[9] J. J. Kanet and V.Sridharan, "Scheduling with inserted idle time: problem taxonomy and literature review", *Operations Research*, vol. 48, 2000, no. 1, pp. 99–110, http: //dx.doi.org/10.1287/opre.48.1.111.12453.

[10] H. Kise, T. Ibaraki and H. Mine, "Performance analysis of six approximation algorithms for the one-machine maximum lateness scheduling problem with ready times", *Journal of the Operations Research Society of Japan,* vol. 22, 1979, pp. 205–224.

[11] J.K. Lenstra, A.H.G. Rinnooy Kan and P.Brucker, "Complexity of machine scheduling problems," *Ann. of Disc. Math.,* 1, 1977, pp. 343–362, http://dx.doi.org/10.1016/s0167-5060(08)707-43-X.

[12] Y. Li, E. Fadda, D. Manerba, R.Tadei and O. Terzo, "Reinforcement Learning Algorithms for Online Single-Machine Scheduling", *Proceedings of the 2020 Federated Conference on Computer Science and Information Systems, M. Ganzha, L. Maciaszek, M. Paprzycki (eds),* ACSIS, vol. 21, 2020, pp. 277–283, http://dx.doi.org/10.15439/2020F100

[13] Z. Liu, "Single machine scheduling to minimize maximum lateness subject to release dates and precedence constraints," *Computers & Operations Research*, vol. 37, 2010, pp. 1537–1543, http://dx.doi.org/10.1016/j.cor.2009.11.008.

[14] E. Nowicki and C. Smutnicki, "An approximation algorithm for a single-machine scheduling problem with release times and delivery times", *Discrete Applied Mathematics,* 48, 1994, pp. 69–79, http://dx.doi.org/10.1016/0166-218X(92)00110-8.

[15] Y. Pan, L. Shi, "Branch and bound algorithm for solving hard instances of the one-machine sequencing problem",*European Journal of Operational Research*, 168, 2006, pp. 1030–1039, http://dx.doi.org/10.1016/j.ejor.2004.07.050.

[16] C.N. Potts, "Analysis of a heuristic for one machine sequencing with release dates and delivery times", *Operations Research*, 28, 1980, pp. 1436–1441, http://dx.doi.org/10.1287/opre.28.6.1436.

[17] L. Schrage, "Optimal Solutions to Resource Constrained Network Scheduling Problems", (unpublished) 1971.

[18] K. Sourirajan and R. Uzsoy, "Hybrid decomposition heuristics for solving large-scale scheduling problems in semiconductor wafer fabrication," *J. Sched.* 10, 2007, pp. 41–65, http://dx.doi.org/10.1007/s10951-006-0325-5.

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 227–230
DOI: 10.15439/2021F93
ISSN 2300-5963 ACSIS, Vol. 25

Group Decision Support for e-Mail Service Optimization through Information Technology Infrastructure Library Framework

Yasen Mitev, Leoneed Kirilov*
Inst. of Information and Communication
Technologies - Bulgarian Academy of Sciences
Acad. G. Bonchev Str., bl. 2, 1113 Sofia, Bulgaria
Email: l_kirilov_8@abv.bg

***Abstract*—The e-mail service takes significant part at the corporate collaboration due to its natural benefits like: unification, traceability and the ease of use. To ensure that such a fundamental service is functioning and being maintained right, proper methods for measuring its efficiency and reliability are in place. In this paper we propose a group decision support that allows the IT Management staff to choose proper asset of key performance indicators (KPIs) for measuring the operational performance of the service in a specific organization. A comprehensive set of KPI indicators is proposed for quality assessment of e-mail service. The optimization of the service is done within ITIL framework.**

***Index Terms*—e-mail service, group decision making, ITIL (Information Technology Infrastructure Library), KPI (Key Performance Indicators)**

I. Introduction

THE e-mail service is one of the business critical functions in most of the enterprises. It is being defined as principal communication channel in most of them. This is caused both by productivity and legal reasons.

The Service Level Agreement is one of the key subjects in the Service Design volume of ITIL (Information Technology Infrastructure Library) - [1]. It is an asset of processes that aims to describe the deliverables that should be achieved in order to have the service available on the expected level. The Key Performance Indicators (KPI) are parameters that quantitatively describe the SLA (Service Level Agreement). For example, when we talk for e-mail service following KPI (Key Performance Indicators) [2] may be defined: 99% of all the e-mail messages to be delivered for less than a minute within the organization; the e-mail servers to be reachable for at least 97,5% of the time; all the priority 1 service requests to be resolved within 90 minutes; etc.

For the scope of this research we are going to cover the KPIs that fall under the Service Operations volume of ITIL - [2]-[4]. Our scenario includes the cases where the e-mail service is already integrated and running in normal operations mode. Key performance indicators can be also used in case of measuring the efficiency of integration of the service or from financial perspective in order to assess the financial efficiency.

II. Problem Formulation and Overview

The usage of ITIL framework for improving and optimizing the level of the email service have been proved as successful approach [2], [5], [6]. This framework does not give the exact rules itself, it also does not specify the exact measurable for success. Therefore, development and application of appropriate methods is actual task – [7]-[14].

In the case study [15] it has been studied how the service is being recognized before and after the ITIL framework implementation based on a simple KPIs defined in [16] and in that way it is shown the benefits and difficulties from implementing ITIL.

How the customer satisfaction is being evaluated is described in [17]. They construct IT service level evaluation system, based on ITIL. On Fig. 1 it can be seen that the KPI has key effect over the customer's perceptions of the quality of the IT Service. The quality of the ITIL service is dependent both by the customer perceptions and the KPIs form the ITIL based service evaluation framework.

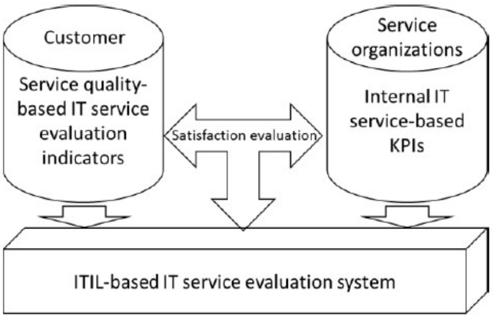

Customer
Service quality-based IT service evaluation indicators
Satisfaction evaluation
Service organizations
Internal IT service-based KPIs
ITIL-based IT service evaluation system

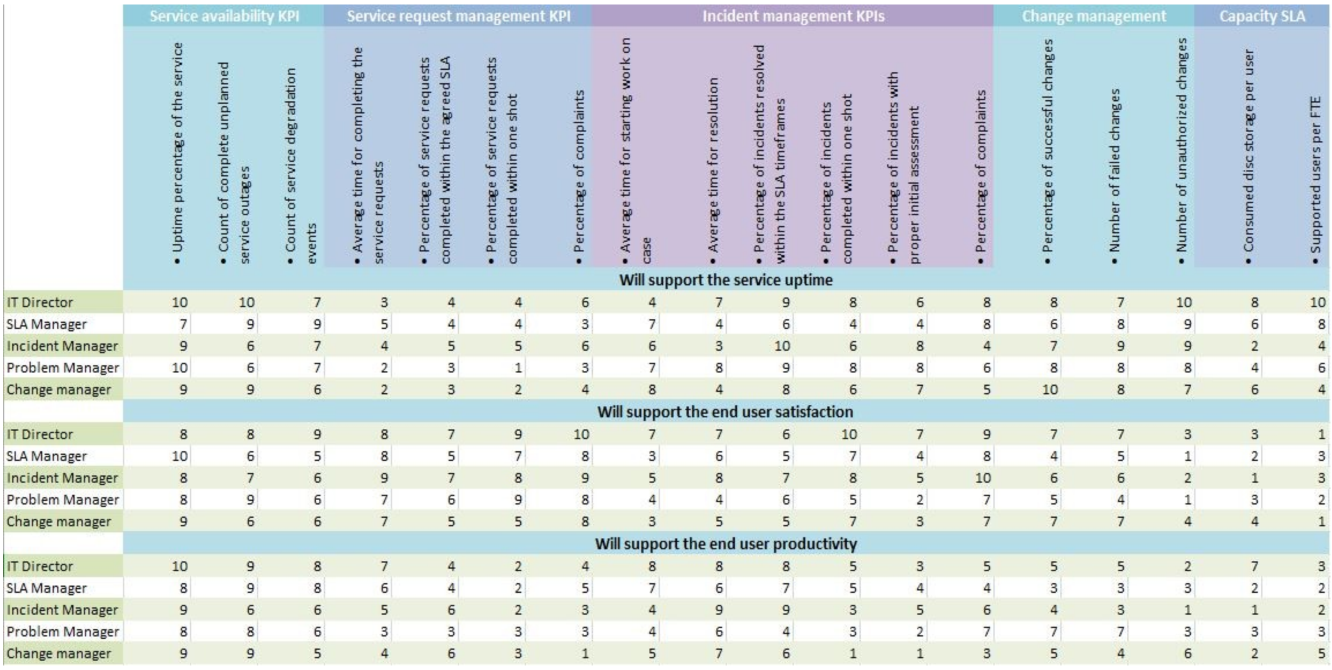
Fig. 1 Satisfaction evaluation for the IT Service - [17]

The proper choosing of the proper KPIs met the following two challenges:

- Usage of the proper asset of KPIs – as there are collaterals suggesting very large lists of KPIs that can describe the properties of the service, it is responsible task to choose the ones that can represent the customer's expectations and priorities. It is a common issue to choose irrelevant KPI metrics that furtherly to be monitored. In that scenario the companies suffer from low customer satisfaction but positive and optimistic values (for example service uptime for more of the expected 98% of the month). This results in losing company resources to get better in tasks that do not add sig-

[*] Corresponding author

TABLE I.
GROUP DECISION MAKING MODEL FOR KPIS



	Service availability KPI			Service request management KPI				Incident management KPIs						Change management			Capacity SLA	
	• Uptime percentage of the service	• Count of complete unplanned service outages	• Count of service degradation events	• Average time for completing the service requests	• Percentage of service requests completed within the agreed SLA	• Percentage of service requests completed within one shot	• Percentage of complaints	• Average time for starting work on case	• Average time for resolution	• Percentage of incidents resolved within the SLA timeframes	• Percentage of incidents completed within one shot	• Percentage of incidents with proper initial assessment	• Percentage of complaints	• Percentage of successful changes	• Number of failed changes	• Number of unauthorized changes	• Consumed disc storage per user	• Supported users per FTE
	Will support the service uptime																	
IT Director	10	10	7	3	4	4	6	4	7	9	8	6	8	8	7	10	8	10
SLA Manager	7	9	9	5	4	4	3	7	4	6	4	4	8	6	8	9	6	8
Incident Manager	9	6	7	4	5	5	6	6	3	10	6	8	4	7	9	9	2	4
Problem Manager	10	6	7	2	3	1	3	7	8	9	8	8	6	8	8	8	4	6
Change manager	9	9	6	2	3	2	4	8	4	8	6	7	5	10	8	7	6	4
	Will support the end user satisfaction																	
IT Director	8	8	9	8	7	9	10	7	7	6	10	7	9	7	7	3	3	1
SLA Manager	10	6	5	8	5	7	8	3	6	5	7	4	8	4	5	1	2	3
Incident Manager	8	7	6	9	7	8	9	5	8	7	8	5	10	6	6	2	1	3
Problem Manager	8	9	6	7	6	9	8	4	4	6	5	2	7	5	4	1	3	2
Change manager	9	6	6	7	5	5	8	3	5	5	7	3	7	7	7	4	4	1
	Will support the end user productivity																	
IT Director	10	9	8	7	4	2	4	8	8	8	5	3	5	5	5	2	7	3
SLA Manager	8	9	8	6	4	2	5	7	6	7	5	4	4	3	3	3	2	2
Incident Manager	9	6	6	5	6	2	3	4	9	9	3	5	6	4	3	1	1	2
Problem Manager	8	8	6	3	3	3	3	4	6	4	3	2	7	7	7	3	3	3
Change manager	9	9	5	4	6	3	1	5	7	6	1	1	3	5	4	6	2	5

nificant value to the overall quality of the service, while other important activities are being neglected.

- Setting up the right values for the chosen KPIs – the most often problem here is that after choosing the KPIs that are going to be monitored, they are not assigned with proper values. This leads to committing with objectives that cannot be met by the service supplier. Also, there is a dependency that determining higher value of a service requires higher effort, materials and funding. Because of that reason choosing the right values has economic dependency as well.

III. KEY PERFORMANCE INDICATORS DESIGN

The design of KPIs is not a single time activity. There are specific occasions when KPIs need to be implemented, applied, followed up and updated. The IT service lifecycle describes the stages where is an interaction with the KPIs.

The basic approaches that are currently being used by the companies are [18]:

- Usage of the proper asset of KPIs – most of the companies rely on a standard asset of KPIs included in their offering plans. These assets are different for the different companies and correspond to their strengths and maturity. This puts demand to the service so it to be relevant only for particular types of business needs. When non-standard requests come to the implementer/developer, custom KPIs needs to be created to measure the bespoke service.
- Setting up the right values over the chosen KPIs – the goal is to determine thresholds for the different KPIs. They need to correspond to the understanding for acceptable quality of service by both customer and supplier. When obtaining these thresholds, a detailed assessment is being made on the available support resource as well as the supported environment. There is also a good practice to include additional warning threshold which to flag that high attention is needed for the indicator in order to continue functioning as expected.

Talking about email service there are a couple of groups with KPIs that can be defined. Depending of the business needs only a couple of the KPI can be chosen and also specific ones to can be added. In some of the companies it is extremely important to have high level of data privacy (banking, military) and in another ones the service reliability and the uptime are the most important (logistics, sales), so in the different business areas there are different business requirement for the email service. That leads to the different usage of KPIs for successfully measuring of the level of the support service.

It is important to be noticed that there are also different groups of KPIs for the different chapters of ITIL - [19].

We propose to use for quality evaluation of e-mail service a number of 18 KPIs divided into the following groups (they are detailed in the header of Table I): *Service availability; Service request management; Incident management; Change management; Capacity SLA*.

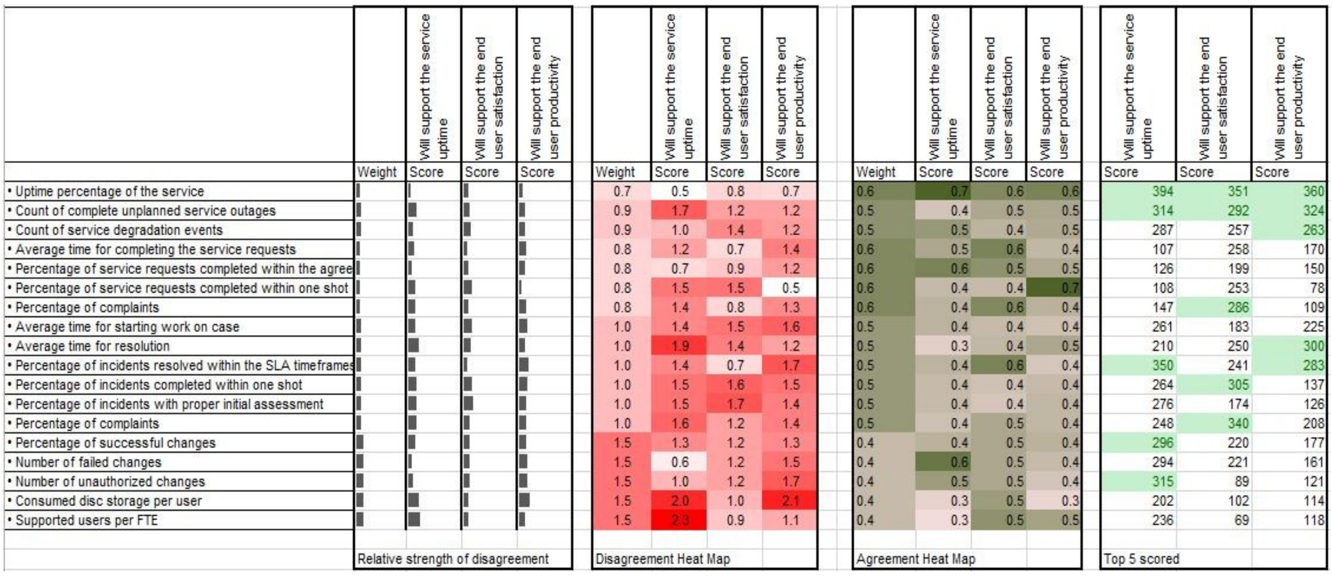
IV. GROUP DECISION MAKING IN KPIS SELECTION

The evaluation and selection of corresponding KPIs is the next step of integrating IT service. We apply Group Decision Making approach for this purpose.

The process is summarized in two steps:

I. The group of experts creates a list with all the key performance indicators that may be included in the SLA for the customer.

TABLE II.
TOP 5 SCORED KPI; LEVEL OF AGREEMENT AND DISAGREEMENT



		Will support the service uptime	Will support the end user satisfaction	Will support the end user productivity		Will support the service uptime	Will support the end user satisfaction	Will support the end user productivity		Will support the service uptime	Will support the end user satisfaction	Will support the end user productivity	Will support the service uptime	Will support the end user satisfaction	Will support the end user productivity
	Weight	Score	Score	Score	Weight	Score	Score	Score	Weight	Score	Score	Score	Score	Score	Score
• Uptime percentage of the service	▮	▮	▮	▮	0.7	0.5	0.8	0.7	0.6	0.7	0.6	0.6	394	351	360
• Count of complete unplanned service outages	▮	▮	▮	▮	0.9	1.7	1.2	1.2	0.5	0.4	0.5	0.5	314	292	324
• Count of service degradation events	▮	▮	▮	▮	0.9	1.0	1.4	1.2	0.5	0.5	0.4	0.5	287	257	263
• Average time for completing the service requests	▮	▮	▮	▮	0.8	1.2	0.7	1.4	0.6	0.5	0.6	0.4	107	258	170
• Percentage of service requests completed within the agree	▮	▮	▮	▮	0.8	0.7	0.9	1.2	0.6	0.6	0.5	0.5	126	199	150
• Percentage of service requests completed within one shot	▮	▮	▮	▮	0.8	1.5	1.5	0.5	0.6	0.4	0.4	0.7	108	253	78
• Percentage of complaints	▮	▮	▮	▮	0.8	1.4	0.8	1.3	0.6	0.4	0.6	0.4	147	286	109
• Average time for starting work on case	▮	▮	▮	▮	1.0	1.4	1.5	1.6	0.5	0.4	0.4	0.4	261	183	225
• Average time for resolution	▮	▮	▮	▮	1.0	1.9	1.4	1.2	0.5	0.3	0.4	0.5	210	250	300
• Percentage of incidents resolved within the SLA timeframes	▮	▮	▮	▮	1.0	1.4	0.7	1.7	0.5	0.4	0.6	0.4	350	241	283
• Percentage of incidents completed within one shot	▮	▮	▮	▮	1.0	1.5	1.6	1.5	0.5	0.4	0.4	0.4	264	305	137
• Percentage of incidents with proper initial assessment	▮	▮	▮	▮	1.0	1.5	1.7	1.4	0.5	0.4	0.4	0.4	276	174	126
• Percentage of complaints	▮	▮	▮	▮	1.0	1.6	1.2	1.4	0.5	0.4	0.5	0.4	248	340	208
• Percentage of successful changes	▮	▮	▮	▮	1.5	1.3	1.2	1.3	0.4	0.4	0.5	0.4	296	220	177
• Number of failed changes	▮	▮	▮	▮	1.5	0.6	1.2	1.5	0.4	0.6	0.5	0.4	294	221	161
• Number of unauthorized changes	▮	▮	▮	▮	1.5	1.0	1.2	1.7	0.4	0.5	0.5	0.4	315	89	121
• Consumed disc storage per user	▮	▮	▮	▮	1.5	2.0	1.0	2.1	0.4	0.3	0.5	0.3	202	102	114
• Supported users per FTE	▮	▮	▮	▮	1.5	2.3	0.9	1.1	0.4	0.3	0.5	0.5	236	69	118
	Relative strength of disagreement				Disagreement Heat Map				Agreement Heat Map				Top 5 scored		

II. The group of experts evaluates the feasibility of the collected KPIs one by one.

We demonstrate the proposed approach on the following real-life problem. There is a need to improve the quality of the IT service in a large national university with ~24 000 students. A number of five experts have been engaged to solve the problem according to the selected KPIs. The experts are part of the university IT department and they have the following roles according to ITIL: IT Director; SLA Manager; Incident Manager; Problem Manager; Change Manager. Additional clarification has been made that the university email service is provided only to the teachers and the personnel and not to the students.

These experts rank each of the indicators with score between 1 and 10 as 1 means that the KPI will not be supportive at all and 10 means that such a KPI will strongly support measuring the organization's performance. Each KPI should be evaluated from 3 aspects – *if it is going to support the service uptime, the user satisfaction and the user productivity.* This will help the process managers to gain clear overview for which purposes the KPIs can be used during the service operations.

The evaluation is made on the base of the IT Environment of the university as follows: technical infrastructure overview; business goals and ongoing issues.

Technical infrastructure overview: The University contains six buildings in one campus, connected with high broadband WAN network in between. The provided e-mail service is available only for the teachers and the administrative personnel. There are 850 mailboxes created in total. The specific is that there is a very large number of external mail contacts stored in the active directory ~32 000. That is due to the reason that for each student a mail contact is created. These contacts are part of different public distribution lists that describe the different classes and learning groups. Technically the environment is hosted on premise in a dedicated server room. Microsoft Exchange 2010 servers with full redundancy deliver the service.

Business goals: It has been planned to upgrade its environment to Exchange 2013 in order to use the features of the latest version. The goal is to have 0% outages for the email service during the weekdays. Another goal is to implement the laboratories booking trough the Exchange calendar feature.

Ongoing issues: currently the personnel is complaining that the support desk is engaged with a big delay after the issue is reported – sometimes on the next business day. Another identified issue is the data loss for email items – a big number of the requested mailbox restores are not successful.

Based on the provided description the experts have put their ratings. Consolidated view of group decision making model can be seen on Table I. The columns correspond to the KPIs. The rows correspond to the DMs: DM1 = IT Director, DM2 = SLA Manager and so on. The values a(i,j) in the matrix are the scores of the Decision Maker (i) according to the KPI(j).

The above model is solved using the group decision support method according to [20]. It provides structured, transparent decision making within a group based on statistical methods. The approach employs a weighted decision matrix with authoritative attributes which leads to an individual decision outcome. The weighting coefficients are used to represent the depth of knowledge for the experts about the area of particular KPI. The solution process consists of three stages: I – Group factor identification; II – Individual scoring; III – Facilitator complies results. The output includes the following data: Disagreement and Agreement heat map; Points of contention; Optimistic/Pessimistic Disagreement;

Optimistic/Pessimistic support of the final score – see Table II.

It can be seen the top 5 scored KPI indicators for each of the three purposes of the feedback session: support for the service uptime; support the end-user satisfaction; support for the end-user productivity. Also, the levels of agreement and disagreement between the experts about the relevancy of particular KPI according to the heat maps are displayed. More intensive color is about high level of agreement (disagreement) between the experts and vice versa.

Further it can be seen that the service availability KPIs (the first three columns from Table I) have major importance for the 3 measured aspects. This is also aligned with high level of agreement between the experts. Also, the level of disagreement between the experts is relatively high for the top 5 chosen KPIs for measuring the end user productivity and satisfaction (Table II, last two columns). That can be explained with the different point of view on the IT service that the different experts have. Another interesting result is that the experts are confident and have high level of agreement for the KPIs that are scored low (see Table II, agreement heat map). That means that we can confidently confirm which KPIs are not relevant. Namely:

- Will support the service uptime: Percentage of service requests completed within the agreed SLA; Average time for completing the service requests; Average time for starting work on case;
- Will support the end user satisfaction: Number of unauthorized changes; Consumed disc storage per user; Supported users per FTE;
- Will support the end user productivity: Percentage of service requests completed within one shot; Supported users per FTE; Percentage of service requests completed within the agreed SLA.

V. Conclusion

We have proposed a methodology for selection of KPIs that to ensure improved client satisfaction. Also, a comprehensive catalogue with Key Performance Indicators for measuring the quality of an email services was presented. Thirdly, a methodology based on group decision making approach for evaluating KPIs relevance is applied. This methodology allows the management department in organizations to have structured approach for choosing proper KPIs for measuring the business goals. The methodology is demonstrated on a real-life example for enhancing the quality of e-mail service in a large organization.

References

[1] L. Hunnebeck, ITIL Service Design, The Stationery Office, London, 2011, ISBN 978-0113313051.

[2] M. Talla and R. Valverde, “An Implementation of ITIL Guidelines for IT Support Process in a Service Organization“, International Journal of Information and Electronics Engineering, Vol.3, No.3, 2013, ISSN: 2010-3719.

[3] R. A. Steinberg, ITIL Service Operation; The Stationery Office; 2011, London; ISBN 978-0113313075.

[4] B. Trinkenreich and G. Santos, G., “Metrics to Support IT Service Maturity Models – A Case Study”, Proceedings of the 17th International Conference on Enterprise Information Systems (ICEIS), (Eds. S. Hammoudi, L. Maciaszek and E. Teniente), vol. 2, p.330-338; 2015, Barcelona, Spain.

[5] W. Guo and Y. Wang, “An Incident Management Model for SaaS Application in the IT Organization.”, Proceedings of the Int. Conf. on Research Challenges in Computer Science-ICRCCS '09, pp. 137-140, 2009 ISBN: 978-0-7695-3927-0.

[6] M. Spremic, Z. Zmirak, K. Kraljevic, “IT and business process performance management: Case study of ITIL implementation in finance service industry.”, Proc. Of the Int. Conf. on Information Technology Interfaces, 2008, 23-26 June 2008, Dubrovnik, pp. 243 – 250.

[7] D. Borissova, “Group decision making for selection of k-best alternatives”, Comptes rendus de l’Acad´emie bulgare des Sciences, 69 (2), 2016, pp. 183-190.

[8] I. Petrov, “On structural entropy and concentration analysis of industrial and market systems.”, In: (Ed. R. Andreev) Proceedings of the Int. Conference on Big Data, Knowledge and Control Systems Engineering 2016, 11-24, Publisher: Union on Automatics and Informatics, Sofia. ISSN: 2367-6450.

[9] Al. Tsenov, “Approaches for Improvement of IT Systems Management”, International Journal of Innovative Science and Modern Engineering (IJISME), 3(6), 2015, pp. 95-98, ISSN: 2319-6386

[10] I. Popchev, Ir. Radeva. and Ir. Nikolova, “Aspects of the evolution from risk management to enterprise global risk management”, Engineering Sciences, LVIII, 2021, No. 1, pp. 16 – 30.

[11] P. Weichbroth, “Mining e-mail message sequences from log data”, Proceedings of the 2018 Federated Conference on Computer Science and Information Systems, M. Ganzha, L. Maciaszek, M. Paprzycki (eds). ACSIS, Vol. 15, pp. 845–848 (2018), DOI: http://dx.doi.org/10.15439/2018F325

[12] L Kirilov, V Guliashki, K Genova, M Vassileva, B Staykov, “Generalized scalarizing model GENS in DSS WebOptim,”, International Journal of Decision Support System Technology (IJDSST) vol. 5, issue 3, 2013, pp. 1-11, DOI: 10.4018/jdsst.2013070101.

[13] J. Rubio and M. Arcilla, “How to Optimize the Implementation of ITIL through a Process Ordering Algorithm”, Applied Sciences, 10 (1), 2020, 34; https://doi.org/10.3390/app10010034

[14] J. L. Rubio Sánchez, "Model to Optimize the Decision Making on Processes in IT Departments" Mathematics 9, no. 9: 983. 2021. https://doi.org/10.3390/math9090983

[15] R. Valverde, R. George, S. Talla, M. Talla, “ITIL-based IT service support process reengineering”, Intelligent Decision Technologies, pp. 1–20, 2013, IDT-130182.

[16] T. Eikebrokk and J. Iden, “Strategising IT service management through ITIL implementation: model and empirical test”, Total Quality Management & Business Excellence, vol. 28, No 3-4, pp. 238-265, 2017.

[17] Y. Xiaozhong, L. Jian and Y. Yong, “Study on the IT Service Evaluation System in ITIL-based Small and Medium-sized Commercial Banks.”, International Journal of Hybrid Information Technology, vol. 8, No.4, pp. 233-242, 2015, ISSN: 1738-9968.

[18] C. Pollard and A. Cater-Steel, “Justifications, strategies, and critical success factors in successful ITIL implementations in US and Australian companies: an exploratory study”, Information systems management, vol. 26, No 2, pp. 164-175, 2009.

[19] M. Brenner, “Classifying ITIL Processes; A Taxonomy under Tool Support Aspects”, Proceedings of the First IEEE/IFIP International Workshop on Business-Driven IT management (BDIM 2006). DOI: 10.1109/BDIM.2006.1649207.

[20] D. Krapohl, “A Structured Methodology for Group Decision Making”, 2012, online http://www.augmentedintel.com/content/articles/group_strategic_decision_making_with_weighted_decision_matrix.asp, (last accessed on: .05.2021)

DOI: 10.15439/2021F121
ISSN 2300-5963 ACSIS, Vol. 25

Combinatorial etude

Miroslav Stoenchev *, Venelin Todorov†‡
*Technical University of Sofia, Bulgaria
†Institute of Mathematics and Informatics, Bulgarian Academy of Sciences
‡Institute of Information and Communication Technologies, Bulgarian Academy of Sciences
Email: mrs@tu-sofia.bg, vtodorov@math.bas.bg, venelin@parallel.bas.bg,

***Abstract*—The purpose of this article is to consider a special class of combinatorial problems, the so called Prouhet-Tarry-Escot problem, solution of which is realized by constructing finite sequences of ± 1. For example, for fixed $p \in \mathbb{N}$, is well known the existence of $n_p \in \mathbb{N}$ with the property: any set of n_p consecutive integers can be divided into 2 sets, with equal sums of its p^{th}-powers. The considered property remains valid also for sets of finite arithmetic progressions of complex numbers.**

I. Morse sequence

For every positive integer m, let us denote with $\vartheta(m)$ and $\varrho(m)$ respectively the number of occurences of digit 1 in the binary representation of m, and the position of first digit 1 in the binary representation of m. The Morse sequence $\{a_m\}_{m=1}^{\infty}$ ([1], [4]) is defined by

$$a_m = (-1)^{\vartheta(m)+\varrho(m)-2}.$$

The following properties are derived directly:

$$a_{2^k} = (-1)^k$$

$$a_{2^k+l} = -a_l \quad \text{for} \quad l = 1, 2 \ldots, 2^k.$$

The problem of finding a number n_p, such that the set $A_{n_p} = \{1, 2, \ldots, n_p\}$ is represented as disjoint union of two subsets, say B and C, with the property:

$$\sum_{b \in B} b^p = \sum_{c \in C} c^p,$$

is solved by the sequence $\{a_m\}_{m=1}^{\infty}$. Elementary proof is given below[1] and $n_p = 2^{p+1}$ has the desired property, with

$$B = \{m \in A_{n_p} : \ a_m = 1\},$$

$$C = A_{n_p} \backslash B = \{m \in A_{n_p} : \ a_m = -1\}.$$

This result can be generalized to arbitrary arithmetic progressions of complex numbers. As example, if $a, d \in \mathbb{C}$, $d \neq 0$ and $A_{n_p} = \{a + kd : \ k = 0, 1, \ldots, n_p - 1\}$, then $n_p = 2^{p+1}$ and $B = \{a + kd \in A_{n_p} : \ a_{k+1} = 1\}$.

II. Formulation of the main results

Let us define $\{H_{n,m}(z)\}_{n,m=1}^{\infty}$, by

$$H_{n,m}(z) = \sum_{l=n}^{\infty} \sum_{k=1}^{2^l} a_k \left(P(z) + k.Q(z)\right)^m,$$

where $P, Q \in \mathbb{C}[z]$ are complex polynomials.

***Proposition** 1:* If $m = 0, 1, \ldots, n-1$ then $H_{n,m} \equiv 0$, while if n is even number the following equality is satisfied

$$H_{n,n}(z) = n! 2^{\frac{n^2-n}{2}} Q^n(z).$$

***Proposition** 2:* Let $n \in \mathbb{N}$ be a even number and $\alpha_1, \alpha_2, \ldots, \alpha_n$ are complex numbers, then

$$\sum_{k=1}^{2^n} a_k(\alpha_1 + k)(\alpha_2 + k) \cdots (\alpha_n + k) = n! 2^{\frac{n^2-n}{2}}.$$

***Proposition** 3:* If $P \in \mathbb{C}[z]$ is a complex polynomial, then

$$\sum_{k=1}^{2^{1+\deg P}} a_k P(k) = 0.$$

***Proposition** 4:* Let p and k be positive integers. Then there exist $n \in \mathbb{N}$, $n \leq 2^{p\lceil \log_2 k \rceil}$ and distinct square-free positive integers x_{ij}, $i = 1, 2, \ldots, k$; $j = 1, 2, \ldots, n$ with the property:

$$\sum_{j=1}^{n} x_{1j}^r = \sum_{j=1}^{n} x_{2j}^r = \cdots = \sum_{j=1}^{n} x_{kj}^r, \ \forall r = 1, 2, \ldots, p.$$

Let $\{b_i\}_{i=0}^{\infty}$ and $\{d_i\}_{i=0}^{\infty}$ be arbitrary sequences of complex numbers. Let γ_n and Γ_n be the sets:

$$\gamma_m = \{b_m + kd_m : \ k \in \mathbb{Z}\} \text{ and}$$

$$\Gamma_n = \gamma_1 \gamma_2 \ldots \gamma_n = \{\prod_{i=1}^{n} (b_i + kd_i) : \ k \in \mathbb{Z} \ \}.$$

***Proposition** 5:* Let n and m be positve integers. Then there exists an integer $s = s(n, m)$ with the property: every $s-$element subset of Γ_n, where k runs through s consequtive integers, can be represented as disjoint union of m subsets, with equal sums of the elements in each one.

The proof of each of the formulated above propositions, with the exception for 5, is based on the following lemma:

[1] Similar solutions and generalizations of the Prouhet-Tarry-Escot problem are considered in [2], [3], [5], [6], [8]

***Lemma** 1:* Set $a, d \in \mathbb{C},\ d \neq 0,\ p \in \mathbb{N}$ and $A_{2^{p+1}} = \{a + kd:\ k = 0, 1, \ldots, 2^{p+1} - 1\}$. Then there are sets $B \cap C = \emptyset$, $B \cup C = A_{2^{p+1}}$ such that

$$\sum_{b\in B} b^p = \sum_{c\in C} c^p.$$

***Corollary** 1:* Under assumptions of lemma 1, it holds

$$\sum_{b\in B} b^r = \sum_{c\in C} c^r,\ r = 0, 1, \ldots, p.$$

To prove lemma 1 and its consequence, we define a sequence of polynomials: $\{T_{s,p}(z)\}_{s=0}^{\infty}$, through which we will gradually calculate the differences between the sums of equal powers of the elements in $B = \{a + kd \in A_{2^{p+1}}:\ a_{k+1} = 1\}$ and $C = \{a + kd \in A_{2^{p+1}}:\ a_{k+1} = -1\}$. For $s \geq 0$ set

$$T_{s,p}(z) = \sum_{k=0}^{4^{s+1}-1} a_{k+1}(z + kd)^p$$

and we calculate

$$T_{s,p}(z) =$$

$$\sum_{0\leq k\leq 4^{s+1}-1; a_{k+1}=1} (z + kd)^p - \sum_{0\leq k\leq 4^{s+1}-1;\ a_{k+1}=-1} (z + kd)^p.$$

When $s \leq \frac{p-1}{2}$, set $z = a$ to obtain

$$T_{s,p}(a) = \sum_{b\leq a+(4^{s+1}-1)d} b^p - \sum_{c\leq a+(4^{s+1}-1)d} c^p,$$

where summation is by $b \in B,\ c \in C$.
Set $p = 2m + r,\ r \in \{0, 1\}$. Here and everywhere below the summations are performed on all $b \in B$ and $c \in C$, which satisfy the corresponding inequalities.
When $r = 1$ we obtain

$$T_{m,p}(a) = \sum_{b\leq a+(4^{m+1}-1)d} b^p - \sum_{c\leq a+(4^{m+1}-1)d} c^p$$

$$= \sum_{b\leq a+(2^{p+1}-1)d} b^p - \sum_{c\leq a+(2^{p+1}-1)d} c^p$$

$$= \sum_{b\in B} b^p - \sum_{c\in C} c^p.$$

When $r = 0$:

$$T_{m-1,p}(a) = \sum_{b\leq a+(2^{2m}-1)d} b^p - \sum_{c\leq a+(2^{2m}-1)d} c^p$$

$$= \sum_{b\leq a+(2^p-1)d} b^p - \sum_{c\leq a+(2^p-1)d} c^p.$$

On the other hand

$$\sum_{a+2^pd\leq b\leq a+(2^{p+1}-1)d} b^p - \sum_{a+2^pd\leq c\leq a+(2^p-1)d} c^p$$

$$= \sum_{2^p\leq m\leq 2^{p+1}-1} a_{m+1}(a + md)^p$$

$$= \sum_{k=0}^{2^p-1} a_{2^p+k+1}\,(a + (2^p + k)d)^p$$

$$= -\sum_{k=0}^{2^p-1} a_{k+1}\,(a + (2^p + k)d)^p$$

$$= -\sum_{k=0}^{2^p-1} a_{k+1}\,((a + 2^pd) + kd)^p =$$

$$= -\sum_{k=0}^{2^{2m}-1} a_{k+1}\,((a + 2^pd) + kd)^p$$

$$= -T_{m-1,p}(a + 2^pd).$$

Therefore, for $p = 2m$ we obtain

$$\sum_{b\in B} b^p - \sum_{c\in C} c^p =$$

$$T_{m-1,p}(a) - T_{m-1,p}(a + 2^pd).$$

Summarized:

$$\sum_{b\in B} b^p - \sum_{c\in C} c^p =$$

$$\begin{cases} T_{m,p}(a), & \text{for} \quad p = 2m + 1 \\ T_{m-1,p}(a) - T_{m-1,p}(a + 2^pd), & \text{for} \quad p = 2m \end{cases}$$

III. Proof of the main results

Lemma 1 follows directly from :
***Proposition** 6:*

$$T_{m-1,p}(z) = \begin{cases} 0, & \text{when} \quad p = 2m - 1 \\ p!2^{\frac{p^2-p}{2}}d^p, & \text{when} \quad p = 2m \end{cases}$$

***Proof** 1:* Let us determine the polynomials $\{T_{s,p}(z)\}_{s=0}^{\infty}$ by finding recurrent formula. Since $a_1 = a_4 = 1,\ a_2 = a_3 = -1$, then

$$T_{0,p}(z) = (z + 3d)^p - (z + 2d)^p - (z + d)^p + z^p.$$

We will prove that for all $s \geq 1$ is valid

$$T_{s,p}(z) = T_{s-1,p}(z + 3.4^sd) - T_{s-1,p}(z + 2.4^sd)$$
$$-T_{s-1,p}(z + 4^sd) + T_{s-1,p}(z).$$

For example, if $s = 1$ then:

$$T_{1,p}(z) = \sum_{k=0}^{15} a_{k+1}(z + kd)^p$$

$$= \sum_{k=0}^{3} a_{k+1}(z + kd)^p + \sum_{k=4}^{7} a_{k+1}(z + kd)^p$$

$$+ \sum_{k=8}^{11} a_{k+1}(z + kd)^p + \sum_{k=12}^{15} a_{k+1}(z + kd)^p$$

$$= T_{0,p}(z) + \sum_{m=0}^{3} a_{2^2+m+1}((z + 4d) + md)^p$$

$$+\sum_{m=0}^{3} a_{2^3+m+1}((z+2.4d)+md)^p$$

$$+\sum_{m=0}^{3} a_{2^3+2^2+m+1}((z+3.4d)+md)^p$$

$$=T_{0,p}(z)-\sum_{m=0}^{3} a_{m+1}((z+4d)+md)^p$$

$$-\sum_{m=0}^{3} a_{m+1}((z+2.4d)+md)^p$$

$$+\sum_{m=0}^{3} a_{m+1}((z+3.4d)+md)^p$$

$$=T_{0,p}(z)-T_{0,p}(z+4d)$$

$$-T_{0,p}(z+2.4d)+T_{0,p}(z+3.4d).$$

The proof is similar in the general case:

$$T_{s,p}(z)=\sum_{k=0}^{4^{s+1}-1} a_{k+1}(z+kd)^p$$

$$=\sum_{k=0}^{4^s-1} a_{k+1}(z+kd)^p+\sum_{k=4^s}^{2.4^s-1} a_{k+1}(z+kd)^p$$

$$+\sum_{k=2.4^s}^{3.4^s-1} a_{k+1}(z+kd)^p+\sum_{k=3.4^s}^{4^{s+1}-1} a_{k+1}(z+kd)^p$$

$$=T_{s-1,p}(z)+\sum_{m=0}^{4^s-1} a_{4^s+m+1}((z+4^sd)+md)^p$$

$$+\sum_{m=0}^{4^s-1} a_{2.4^s+m+1}((z+2.4^sd)+md)^p$$

$$+\sum_{m=0}^{4^s-1} a_{3.4^s+m+1}((z+3.4^sd)+md)^p$$

$$=T_{s-1,p}(z+3.4^sd)-T_{s-1,p}(z+2.4^sd)$$

$$-T_{s-1,p}(z+4^sd)+T_{s-1,p}(z),$$

whence the necessary recurrent formula is established.

In the case $1\leq s\leq\left[\frac{p}{2}\right]-1$, we prove that $T_{s,p}(z)$ has the type:

$$T_{s,p}(z)=\sum_{i_1=2s}^{p-2}\sum_{i_2=2(s-1)}^{i_1-2}\sum_{i_3=2(s-2)}^{i_2-2}\cdots$$

$$\cdots\sum_{i_{s+1}=0}^{i_s-2}\binom{p}{i_1}\binom{i_1}{i_2}\binom{i_2}{i_3}\cdots\binom{i_s}{i_{s+1}}d^{p-i_{s+1}}L_{s,p}z^{i_{s+1}},$$

where

$$L_{s,p}=(3^{p-i_1}-2^{p-i_1}-1)(3^{i_1-i_2}-2^{i_1-i_2}-1)\cdots$$

$$\cdots(3^{i_s-i_{s+1}}-2^{i_s-i_{s+1}}-1)4^{i_1+i_2+\cdots+i_s-si_{s+1}}.$$

Indeed, when $s=0$ follows:

$$T_{0,p}(z)=(z+3d)^p-(z+2d)^p-(z+d)^p+z^p$$

$$=\sum_{i_1=0}^{p-2}\binom{p}{i_1}(3^{p-i_1}-2^{p-i_1}-1)d^{p-i_1}z^{i_1}.$$

Direct calculation for $T_{1,p}$ gives

$$T_{1,p}(z)=T_{0,p}(z)-T_{0,p}(z+4d)$$

$$-T_{0,p}(z+2.4d)+T_{0,p}(z+3.4d)$$

$$=\sum_{i_1=0}^{p-2}\binom{p}{i_1}(3^{p-i_1}-2^{p-i_1}-1)d^{p-i_1}$$

$$\big((z+12d)^{i_1}-(z+8d)^{i_1}-(z+4d)^{i_1}+z^{i_1}\big)$$

$$=\sum_{i_1=2}^{p-2}\binom{p}{i_1}(3^{p-i_1}-2^{p-i_1}-1)d^{p-i_1}$$

$$\big((z+12d)^{i_1}-(z+8d)^{i_1}-(z+4d)^{i_1}+z^{i_1}\big)$$

$$=\sum_{i_1=2}^{p-2}\binom{p}{i_1}(3^{p-i_1}-2^{p-i_1}-1)d^{p-i_1}$$

$$\left(z^{i_1}+\sum_{i_2=0}^{i_1}\binom{i_1}{i_2}(3^{i_1-i_2}-2^{i_1-i_2}-1)(4d)^{i_1-i_2}z^{i_2}\right)$$

$$=\sum_{i_1=2}^{p-2}\binom{p}{i_1}(3^{p-i_1}-2^{p-i_1}-1)d^{p-i_1}$$

$$\left(\sum_{i_2=0}^{i_1-2}\binom{i_1}{i_2}(3^{i_1-i_2}-2^{i_1-i_2}-1)(4d)^{i_1-i_2}z^{i_2}\right)=$$

$$=\sum_{i_1=2}^{p-2}\sum_{i_2=0}^{i_1-2}\binom{p}{i_1}\binom{i_1}{i_2}(3^{p-i_1}-2^{p-i_1}-1)$$

$$(3^{i_1-i_2}-2^{i_1-i_2}-1)4^{i_1-i_2}d^{p-i_2}z^{i_2}=$$

$$=\sum_{i_1=2}^{p-2}\sum_{i_2=0}^{i_1-2}\binom{p}{i_1}\binom{i_1}{i_2}d^{p-i_2}L_{1,p}z^{i_2},$$

hence the assertion is established for $s=1$. Suppose that for some $s\geq 2$, $T_{s-1,p}(z)$ satisfies the recurent formula and denote

$$G^{s,p}_{i_1,i_2,\ldots,i_{s+1}}=\binom{p}{i_1}\binom{i_1}{i_2}\binom{i_2}{i_3}\cdots\binom{i_s}{i_{s+1}}d^{p-i_{s+1}}L_{s,p},$$

for $s\geq 1$. Direct calculation shows:

$$T_{s,p}(z)=T_{s-1,p}(z+3.4^sd)-T_{s-1,p}(z+2.4^sd)$$

$$-T_{s-1,p}(z+4^sd)+T_{s-1,p}(z)$$

$$=\sum_{i_1=2(s-1)}^{p-2}\sum_{i_2=2(s-2)}^{i_1-2}\sum_{i_3=2(s-3)}^{i_2-2}\cdots\sum_{i_s=0}^{i_{s-1}-2}G^{s-1,p}_{i_1,i_2,\ldots,i_s}$$

$$((z+3.4^s d)^{i_s}-(z+2.4^s d)^{i_s}-(z+4^s d)^{i_s}+z^{i_s})$$

$$=\sum_{i_1=2(s-1)}^{p-2}\sum_{i_2=2(s-2)}^{i_1-2}\cdots\sum_{i_s=0}^{i_{s-1}-2}G^{s-1,p}_{i_1,i_2,\ldots,i_s}$$

$$[z^{i_s}+\sum_{i_{s+1}=0}^{i_s}\binom{i_s}{i_{s+1}}(3^{i_s-i_{s+1}}-2^{i_s-i_{s+1}}-1)$$

$$4^{s(i_s-i_{s+1})}d^{i_s-i_{s+1}}z^{i_{s+1}}]$$

$$=\sum_{i_1=2s}^{p-2}\sum_{i_2=2(s-1)}^{i_1-2}\cdots\sum_{i_s=2}^{i_{s-1}-2}G^{s-1,p}_{i_1,i_2,\ldots,i_s}\sum_{i_{s+1}=0}^{i_s-2}$$

$$\binom{i_s}{i_{s+1}}(3^{i_s-i_{s+1}}-2^{i_s-i_{s+1}}-1)4^{s(i_s-i_{s+1})}d^{i_s-i_{s+1}}z^{i_{s+1}}$$

$$=\sum_{i_1=2s}^{p-2}\sum_{i_2=2(s-1)}^{i_1-2}\cdots\sum_{i_s=2}^{i_{s-1}-2}\sum_{i_{s+1}=0}^{i_s-2}G^{s-1,p}_{i_1,i_2,\ldots,i_s}\binom{i_s}{i_{s+1}}$$

$$(3^{i_s-i_{s+1}}-2^{i_s-i_{s+1}}-1)4^{s(i_s-i_{s+1})}d^{i_s-i_{s+1}}z^{i_{s+1}}$$

$$=\sum_{i_1=2s}^{p-2}\sum_{i_2=2(s-1)}^{i_1-2}\cdots\sum_{i_s=2}^{i_{s-1}-2}\sum_{i_{s+1}=0}^{i_s-2}G^{s,p}_{i_1,i_2,\ldots,i_{s+1}}z^{i_{s+1}},$$

which prove that $T_{s,p}(z)$ satisfies the recurrent formula.

Let us determine the degree of $T_{s,p}(z),\ s\geq 0$. According to the derived formula we find $i_{s+1}\leq i_s-2\leq i_{s-1}-4\leq\cdots\leq i_1-2s\leq p-2(s+1)$, as equality is reached everywhere. Therefore $\deg T_{s,p}(z)=p-2(s+1)$. If $p=2m+r,\ r\in\{0,1\}$, then

$$\deg T_{m-1,p}(z)=p-2m=r.$$

For $r=0$ we obtain that $T_{m-1,p}(z)$ is a constant, equal to $p!2^{\frac{p^2-p}{2}}d^p$. Indeed

$$T_{m-1,p}(z)=$$

$$=\sum_{i_1=2(m-1)}^{p-2}\sum_{i_2=2(m-2)}^{i_1-2}\cdots\sum_{i_{m-1}=2}^{i_{m-2}-2}\sum_{i_m=0}^{i_{m-1}-2}G^{m-1,p}_{i_1,i_2,\ldots,i_{s+1}}z^{i_m}$$

$$=\sum_{i_1=2(m-1)}^{2(m-1)}\sum_{i_2=2(m-2)}^{2(m-2)}\cdots\sum_{i_{m-1}=2}^{2}\sum_{i_m=0}^{0}$$

$$G^{m-1,p}_{i_1,i_2,\ldots,i_{s+1}}z^{i_m}$$

$$=G^{m-1,p}_{p-2,p-4,p-6\ldots,2,0}$$

$$=\binom{p}{p-2}\binom{p-2}{p-4}\cdots\binom{4}{2}\binom{2}{0}d^pL_{m-1,p}$$

$$=\frac{p!d^p}{2^m}L_{m-1,p}=p!2^{\frac{p^2-p}{2}}d^p\Longrightarrow$$

$$\Longrightarrow T_{m-1,p}(z)=p!2^{\frac{p^2-p}{2}}d^p,\ \text{for } p=2m.$$

In the case $r=1$, we will prove that $T_{m,p}(z)=0$:

$$T_{m,p}(z)=$$

$$=T_{m-1,p}(z+3.4^m d)-T_{m-1,p}(z+2.4^m d)-$$

$$T_{m-1,p}(z+4^m d)+T_{m-1,p}(z)$$

$$=\sum_{i_1=2(m-1)}^{p-2}\sum_{i_2=2(m-2)}^{i_1-2}\cdots\sum_{i_m=0}^{i_{m-1}-2}G^{m-1,p}_{i_1,i_2,\ldots,i_m}$$

$$((z+3.4^m d)^{i_m}-(z+2.4^m d)^{i_m}-(z+4^m d)^{i_m}+z^{i_m})=0,$$

and the above equality is valid, since the summation index i_m takes values 0 and 1. Thus the proposition 6 is proved. Proofs of propositions 1,2,3,4,5 will be presented in [7].

Acknowledgement

Venelin Todorov is supported by the Bulgarian National Science Fund under Project DN 12/5-2017 ”Efficient Stochastic Methods and Algorithms for Large-Scale Problems” and by the National Scientific Program ”Information and Communication Technologies for a Single Digital Market in Science, Education and Security” (ICTinSES), contract No. D01-205/23.11.2018, financed by the Ministry of Education and Science. Miroslav Stoenchev is supported by KP-06-M32/2-17.12.2019 ”Advanced Stochastic and Deterministic Approaches for Large-Scale Problems of Computational Mathematics”.

References

[1] J. -P. Allouche, J. O. Shallit, The ubiquitous Prouhet-Thue-Morse sequence. In C. Ding, T. Helleseth, and H. Niederreiter, editors, Sequences and Their Applications, Proceedings of SETA ’98, pp. 1–16, Springer-Verlag, 1999.

[2] P. Borwein, C. Ingalls, The Prouhet-Tarry-Escott Problem revisited. Enseign. Math. 40 (1994), 3–27. MR 95d:11038

[3] P. Borwein, Computational Excursions in Analysis and Number Theory, Springer-Verlag, New York, 2002.

[4] J. Byszewski and M. Ulas, Some identitites involving the Prouhet-Thue-Morse sequence and its relatives, Acta Math. Hungar., 127-2,438-456, 2015

[5] H.L. Dorwart and O.E. Brown, The Tarry-Escott Problem, M.A.A. Monthly, 44, 613-626, 1937

[6] G.H. Hardy, E.M. Wright, An Introduction to the Theory of Numbers, fifth ed., Oxford Univ. Press, New York,1979

[7] M. Stoenchev, V. Todorov, Combinatorial etudes and number theory, (in appear)

[8] T. Wakhare, C. Vignat, Settling some sum suppositions, arXiv:1805.10569 [math.NT]

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 235–238

DOI: 10.15439/2021F84
ISSN 2300-5963 ACSIS, Vol. 25

An Optimized Technique for Wigner Kernel Estimation

Venelin Todorov
Bulgarian Academy of Sciences
Institute of Mathematics and Informatics
ul. G. Bonchev 8, 1113 Sofia, Bulgaria
Bulgarian Academy of Sciences
Institute of Information and Communication Technologies
ul. G. Bonchev 25A, 1113 Sofia, Bulgaria
Email: vtodorov@math.bas.bg,venelin@parallel.bas.bg

Stefka Fidanova
Bulgarian Academy of Sciences
Institute of Information and Communication Technologies
ul. G. Bonchev 25A, 1113 Sofia, Bulgria
Email: stefka@parallel.bas.bg

Ivan Dimov
Bulgarian Academy of Sciences
Institute of Information and Communication Technologies
ul. G. Bonchev 25A, 1113 Sofia, Bulgria
Email: ivdimov@bas.bg

Stoyan Poryazov
Bulgarian Academy of Sciences
Institute of Mathematics and Informatics
ul. G. Bonchev 8, 1113 Sofia, Bulgaria
Bulgarian Academy of Sciences
Email: stoyan@math.bas.bg

***Abstract*—We study an optimized Adaptive Monte Carlo algorithm for the Wigner kernel - an important problem in quantum mechanics. We will compare the results with the basic adaptive approach and other stochastic approaches for computing the Wigner kernel represented by difficult multidimensional integrals in dimension d up to 12. The higher cases $d > 12$ will be considered for the first time. A comprehensive study and an analysis of the computational complexity of the optimized Adaptive MC algorithm under consideration has also been presented.**

I. Introduction

DIFFERENT mathematical formulations of quantum mechanics exist, among which the ones suggested by E. Schrodinger, E. Wigner, R. Feynman, L.V. Keldysh, K. Husimi, D. Bohm are more frequently used nowadays [10]. The Wigner formulation of quantum mechanics allows the comprehension and prediction of quantum mechanical phenomena in terms of quasidistribution functions. One of the best known physicist Richard Feynman formulated the problem of finding an effective and fast algorithm with linear or polynomial computational complexity for computing multidimensional integrals that represent Wigner kernel [6]. More information about the signed particle formulation of a single-body and many-body system can be found in [8], [9], [10]. Up to now the Wigner kernel is calculated with deterministic methods [11], [12], [14] which suffer from the „curse of dimensionality" [2]. In our previous work [13] we consider the case when the dimension $d \leq 9$, the higher cases $d > 12$ will be considered for the first time in the presented work.

II. Description of the optimized Adaptive approach

Adaptive strategy [1], [3], [4] is well known method for evaluation of multidimensional integrals, especially when the integrand function has peculiarities and peaks. Let p_j and I_{Ω_j} are the following expressions: $p_j = \int_{\Omega_j} p(\mathbf{x})\,\mathrm{dx}$ and $I_{\Omega_j} = \int_{\Omega_j} f(\mathbf{x})p(\mathbf{x})\,\mathrm{dx}$. Consider now a random point $\xi^{(j)} \in \Omega_j$ with a density function $p(\mathbf{x})/p_j$. In this case $I_{\Omega_j} = \mathbf{E}\left[\frac{p_j}{N}\sum_{i=1}^{N} f(\xi_i^{(j)})\right] = \mathbf{E}\theta_N$. This adaptive algorithm gives an approximation with an error $\varepsilon \leq c\ N^{-1/2}$, where $c \leq 0.6745\sigma(\theta)$ ($\sigma(\theta)$ is the standard deviation).

The optimized adaptive algorithm has higher accuracy than the original Adaptive Monte Carlo algorithm as can be seen from the tables below. The increase of the constant for the initial number of taken subregions $M = 8$ improves the relative error compared with the previous choice $M = 2$ in [13]. The optimized adaptive algorithm is described below.

Algorithm

1. **Input data**: *total number of points* $N1$, *constant* $M = 8$*(the initial number of subregions taken), constant* ε *(max value of the variance in each subregion), constant* δ *(maximal admissible number of subregions), d-dimensionality of the initial region/domain,* f *- the function of interest.*

Venelin Todorov is supported by the Bulgarian National Science Fund under Project KP-06-M32/2 - 17.12.2019 "Advanced Stochastic and Deterministic Approaches for Large-Scale Problems of Computational Mathematics" and by the National Scientific Program "Information and Communication Technologies for a Single Digital Market in Science, Education and Security (ICT in SES)", contract No DO1-205/23.11.2018, financed by the Ministry of Education and Science in Bulgaria. The work is also supported by Bulgarian National Science Fund under Project DN 12/5-2017 "Efficient Stochastic Methods and Algorithms for Large-Scale Problems" and by the Project KP-06-Russia/17 "New Highly Efficient Stochastic Simulation Methods and Applications" funded by the National Science Fund - Bulgaria.

IEEE Catalog Number: CFP2185N-ART ©2021, PTI

```
    1.1. **Calculate** the number of points to be taken in each subregion $N = N1/\delta$.
2. **For** $j = 1,\ M^d$:
    2.1. **Calculate** the approximation of $I_{\Omega_j}$ and the variance $\mathbf{D}_{\Omega_j}$ in subdomain $\Omega_j$ based on $N$ independent realizations of random variable $\theta_N$;
    2.2. **If** $(\mathbf{D}_{\Omega_j} \geq \varepsilon)$ **then**
        2.2.1. **Choose** the axis direction on which the partition will perform,
        2.2.2. **Divide** the current domain into two $(G_{j_1}, G_{j_2})$ along the chosen direction,
        2.2.3. **If** the length of obtained subinterval is less than $\delta$ **then go to** step 2.2.1 **else** $j = j_1$ $G_{j_1}$ is the current domain right and **go to** step 2.1;
    2.3. **Else if** $(\mathbf{D}_{\Omega_j} < \varepsilon)$ but an approximation of $I_{G_{j_2}}$ has not been calculated yet, **then** $j = j_2$ $G_{j_2}$ is the current domain along the corresponding direction right and **go to** step 2.1;
    2.4. **Else if** $(\mathbf{D}_{\Omega_j} < \varepsilon)$ but there are subdomains along the other axis directions, **then go to** step 2.1;
    2.5. **Else** Accumulation in the approximation $I_N$ of $I$.
```

Computational complexity

For the simple case when we have the two dimensional case ($N = 2$) and on the first step in the optimized adaptive approach we have $M = 8$ subdomains in our optimized Adaptive approach and

$$\hat{\theta}_N = \sum_{j=1}^{M} \frac{1}{N_j} \sum_{i=j}^{N_M} \theta_i$$

where $\sum\limits_{j=1}^{M} N_j = N$, so we have the same number of operations as the Crude Monte Carlo, which computational complexity is linear [2], to evaluate an approximation of I_{G_j}.

So we choose only $\mathcal{O}(1)$ subdomains where the variance is greater than the parameter ε and this is independent of N. When we divide the domain on every step adaptiveness is not in all subdomains, but only in $\mathcal{O}(1)$ subdomains. At the beginning we have to choose $\frac{N}{k_0}$ random points. After that when dividing the domain into 2^N subdomains, we choose only $\mathcal{O}(1)$ subdomains, this choice is again independent of N. In these subdomains we choose $\frac{N}{k_1}$ points. On the j^{th} step of the Adaptive approach we choose $\mathcal{O}(1)$ subdomains with $\frac{N}{k_j}$ points. We have that $\sum\limits_{j=0}^{i} \frac{1}{k_j} = 1$. Therefore for the computational complexity we obtain

$$\frac{N}{k_0} + \mathcal{O}(1)\frac{N}{k_1} + \cdots + \mathcal{O}(1)\frac{N}{k_i} =$$

$$= N\mathcal{O}(1)\left(\sum_{j=0}^{i} \frac{1}{k_j}\right) = N\mathcal{O}(1) = \mathcal{O}(N).$$

In this way we can conclude that the computational complexity of the optimized Adaptive algorithm is linear.

III. Numerical examples

A new formulation of quantum mechanics in terms of signed classical field-less particles is presented in [7], [10]. Just for completeness we give here the three postulates which completely define the new mathematical formulation of quantum mechanics taken from [7].

Postulate I. Physical systems can be described by means of (virtual) Newtonian particles, i.e. provided with a position $\mathbf{x}$ and a momentum $\mathbf{p}$ simultaneously, which carry a sign which can be positive or negative.

Postulate II. A signed particle, evolving in a potential $V = V(x)$, behaves as a field-less classical point-particle which, during the time interval $\mathrm{d}t$, creates a new pair of signed particles with a probability $\gamma(\mathbf{x}(t))\mathrm{d}t$, where

$$\gamma(\mathbf{x}) = \int_{-\infty}^{+\infty} \mathrm{D}\mathbf{p}' V_W^+(\mathbf{x}; \mathbf{p}') \equiv \lim_{\triangle\mathbf{p}' \to 0^+} \sum_{\mathbf{M}=-\infty}^{+\infty} V_W^+(\mathbf{x}; \mathbf{M}\triangle\mathbf{p}'),$$

where $\hbar = \frac{h}{2\pi}$ is the reduced Planck constant (h) or Dirac constant, $\mathbf{M} = (M_1, M_2, \ldots, M_d)$ is a set of d integers and $V_W^+(\mathbf{x}; \mathbf{p})$ is the positive part of the quantity

$$V_W(\mathbf{x}; \mathbf{p}) = \frac{i}{\pi^d \hbar^{d+1}} \int_{-\infty}^{+\infty} \mathrm{d}\mathbf{x}' e^{-\frac{2i}{\hbar}\mathbf{x}'\mathbf{p}} [V(\mathbf{x}+\mathbf{x}') - V(\mathbf{x}-\mathbf{x}')], \tag{1}$$

known as the Wigner kernel (in a d-dimensional space) [15]. If, at the moment of creation, the parent particle has sign s, position $\mathbf{x}$ and momentum $\mathbf{p}$, the new particles are both located in $\mathbf{x}$, have signs $+s$ and $-s$, and momentum $\mathbf{p} + \mathbf{p}'$ and $\mathbf{p} - \mathbf{p}'$ respectively, with $\mathbf{p}'$ chosen randomly according to the (normalized) probability $\frac{V_W^+(\mathbf{x};\mathbf{p})}{\gamma(\mathbf{x})}$.

Postulate III. Two particles with opposite sign and same phase-space coordinates $(\mathbf{x}, \mathbf{p})$ annihilate.

The infinite domain of integration can be mapped into the s-dimensional unit hypercube using the following transformation $\frac{1}{2} + \frac{1}{\pi}\arctan(x)$ which maps $(-\infty, \infty)$ to $(0, 1)$. We want to compute (1) in the $3, 6, 9$ and for the first time in 12-dimensional case,

$$Vw(x,p) = \int e^{\left(\frac{-i2 \sum\limits_{k=1}^{n} x'_k p_k}{\hbar}\right)} \times$$

$$[V(x_1+x'_1, \ldots x_n+x'_n) - V(x_1-x'_1, \ldots x_n-x'_n)] dx'_1 \ldots dx'_n,$$

where the Wigner potential is $V = V(x) = \{x_1 \ldots x_n,\ \ x', x, p, x + x', x - x' \ \in\ [0,1]\}$. It is well known that Wigner kernel has real values [15].

First, we will make a comparison with deterministic method of mid rectangulars, and after that with the well known

Table I
RELATIVE ERROR OF THE OPTIMIZED ADAPTIVE APPROACH, ADAPTIVE APPROACH AND THE DETERMINISTIC MID RECTANGULAR METHOD

s	N	determ.	t (s)	OptAdapt	t (s)	Adapt	t (s)
3	$32^2 \times 50$	8.51e-03	0.2	1.47e-03	0.1	2.71e-03	0.1
	$32^2 \times 100$	8.21e-03	0.5	8.84e-05	0.31	3.42e-04	0.2
	$64^2 \times 50$	5.76e-03	1	2.32e-05	0.8	7.52e-05	0.55
	$64^2 \times 100$	4.89e-03	1.9	4.23e-06	1.7	1.21e-05	1.3
6	$8^4 \times 50^2$	1.16e-02	41.2	8.64e-05	30	9.09e-04	18.1
	$8^4 \times 100^2$	9.75e-03	160.6	2.31e-06	98	1.52e-05	57.9
	$16^4 \times 50^2$	7.84e-03	635.2	1.53e-05	487	4.37e-04	311.5
	$16^4 \times 100^2$	2.12e-03	2469.1	1.08e-05	1657	3.80e-04	987.1
9	$6^6 \times 16^3$	1.75e-03	835.5	2.66e-05	504	7.62e-05	330.5
	$6^6 \times 32^3$	1.35e-03	5544.1	8.66e-06	3451	2.73e-05	2225.1
	$6^6 \times 40^3$	1.12e-03	10684.4	7.83e-07	6531	8.12e-06	4491.5

stochastic approaches of Sobol QMC and Fibonacci based lattice rule FIBO, see [13].

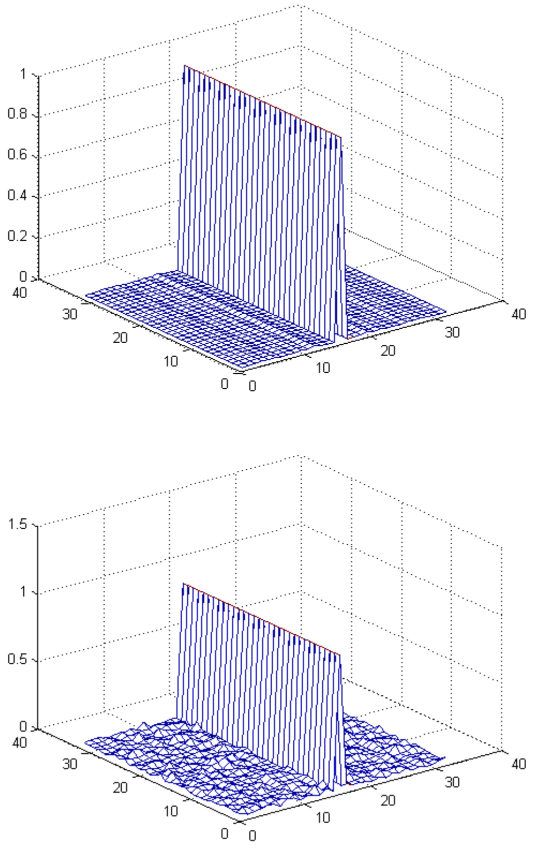


Figure 1. The Wigner kernel with optimized adaptive and standard adaptive method

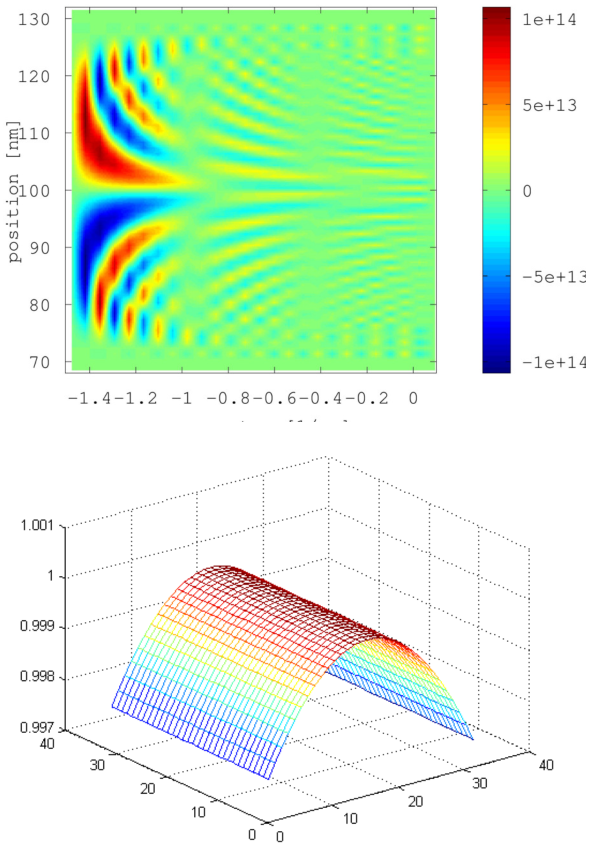


Figure 2. The position and the peak of the Wigner kernel with optimizes adaptive approach

Table II
RELATIVE ERROR FOR 3 DIMENSION

N	Adapt	OptAdapt	FIBO	Sobol
10^3	5.36e-03	3.21e-04	3.72e-02	1.07e-02
10^4	4.84e-04	4.02e-05	7.06e-03	8.77e-03
10^5	2.51e-05	3.33e-06	3.40e-03	8.57e-04
10^6	1.76e-05	1.56e-07	1.01e-03	6.73e-04
10^7	6.26e-06	5.67e-08	1.80e-04	5.98e-05

Table III
RELATIVE ERROR FOR 6 DIMENSION

N	Adapt	OptAdapt	FIBO	Sobol
10^3	6.72e-03	1.07e-04	7.82e-03	2.42e-02
10^4	9.10e-04	2.11e-05	5.01e-03	5.02e-03
10^5	5.26e-05	3.02e-06	6.88e-03	4.60e-04
10^6	2.70e-06	3.21e-07	7.68e-04	3.59e-04
10^7	1.03e-06	5.13e-08	4.12e-04	8.11e-05

In Table I it can be seen that the optimized stochastic approach gives better results and lower relative errors than the adaptive approach used in our previous study [13]. It can be seen that the computational time for the optimized Adaptive

Table IV
RELATIVE ERROR FOR 9 DIMENSION

N	Adapt	OptAdapt	FIBO	Sobol
10^3	4.92e-02	6.11e-04	2.03e-02	5.42e-02
10^4	9.09e-04	1.12e-05	2.02e-03	6.02e-03
10^5	3.32e-05	9.88e-07	9.16e-04	3.57e-03
10^6	6.46e-06	2.01e-07	7.13e-04	8.02e-04
10^7	1.21e-06	5.84e-08	4.84e-04	5.19e-04

Table V
RELATIVE ERROR FOR 12 DIMENSION

N	Adapt	OptAdapt	FIBO	Sobol
10^3	3.91e-03	1.11e-04	1.33e-02	2.85e-02
10^4	5.04e-04	9.01e-06	1.34e-03	4.04e-03
10^5	2.76e-04	3.14e-06	5.51e-04	1.77e-03
10^6	4.14e-05	1.12e-07	4.43e-04	4.07e-04
10^7	2.31e-06	2.83e-08	2.5684e-04	2.7e-04

Table VI
RELATIVE ERROR FOR 15 DIMENSION

t,s	Adapt	OptAdapt	FIBO	Sobol
10^3	6.21e-03	6.22e-04	8.76e-04	2.31e-02
10^4	4.45e-04	4.51e-05	5.56e-04	5.45e-03
10^5	5.43e-05	3.56e-06	3.34e-04	4.11e-03
10^6	1.23e-05	4.16e-07	1.34e-04	6.45e-04

Table VII
RELATIVE ERROR FOR 18 DIMENSION

t,s	Adapt	OptAdapt	FIBO	Sobol
10^3	8.32e-03	1.73e-03	1.42e-03	5.32e-02
10^4	1.05e-03	8.05e-05	7.33e-04	6.31e-03
10^5	2.42e-03	6.32e-06	7.42e-04	4.73e-03
10^6	5.45e-05	7.58e-07	2.71e-04	5.54e-04

MC approach is better than the deterministic method when the dimensionality increases. The advantage of the optimized adaptive algorithm in comparison with the previously used adaptive algorithm is shown on Figure 1, and the computation of the position of the signs and the peak are given in Figure 2. The numerical results including relative errors and computational times corresponding to the algorithms under consideration are presented, and the algorithms efficiency is discussed. A numerical comparison for a given number of samples between the adaptive approach (Adapt) used in [13], the Sobol (Sob) and the Lattice sequences FIBO described in [13] and the new optimized Adaptive approach (OptAdapt) has been given in Tables II-VII. From the all experiments it can be clearly seen that the optimized adaptive approach gives relative errors with at least 1 or 2 orders better than those produced by the adaptive approach, because of the increased number of subregions taken in every subdomain M. The adaptive approach itself gives superior results to the other two stochastic approaches as it is completely described in our previous study [13]. The optimized Adaptive MC approach outperforms the other two approaches FIBO and Sobol QMC by at least 3 orders even for 18 dimensional case, see Table VII. We should emphasize here that the efficiency of the optimized adaptive MC algorithm under consideration is high when computational peculiarities of the integrand occur only in comparatively small subregion of the initial integration domain as it is in the case of the Wigner kernel.

IV. CONCLUSIONS

The optimized adaptive Monte Carlo algorithm under consideration gives the most accurate results in computing the Wigner kernel by a stochastic approach and it has lower computational complexity than the existing deterministic approaches. This means that the proposed optimized stochastic approach is of great importance for the problems in quantum mechanics with high dimensions. Therefore, the presented optimized adaptive MC algorithm is one new successful solution (in terms of robustness and reliability) of Richard Feynman's problem for Wigner kernel evaluation for dimension $d > 12$.

REFERENCES

[1] Berntsen J., Espelid T.O., Genz A. (1991) An adaptive algorithm for the approximate calculation of multiple integrals, ACM Trans. Math. Softw. 17: 437–451.

[2] Dimov I. (2008) Monte Carlo Methods for Applied Scientists, New Jersey, London, Singapore, World Scientific, 291 p., ISBN-10 981-02-2329-3.

[3] Dimov I., Karaivanova A., Georgieva R., Ivanovska S. (2003) Parallel Importance Separation and Adaptive Monte Carlo Algorithms for Multiple Integrals, Springer Lecture Notes in Computer Science, 2542, 99–107.

[4] Dimov I., Georgieva R. (2010) Monte Carlo Algorithms for Evaluating Sobol' Sensitivity Indices. Math. Comput. Simul. 81(3): 506–514.

[5] Ermakov S.M. (1985) Monte Carlo Methods and Mixed Problems, Nauka, Moscow.

[6] Feynman R.P. (1948) Space-time approach to non-relativistic quantum mechanics, Rev. Mod. Phys. 20.

[7] Sellier J.M. (2015) A signed particle formulation of non-relativistic quantum mechanics, Journal of Computational Physics 297: 254—265.

[8] Sellier J.M., Dimov I. (2016) On a full Monte Carlo approach to quantum mechanics, Physica A: Statistical Mechanics and its Applications Volume 463: 45–62.

[9] Sellier J.M., Dimov I. (2014) The many-body Wigner Monte Carlo method for time-dependent ab-initio quantum simulations, J. Comput. Phys. 273: 589–597.

[10] Sellier J.M., Nedjalkov M., Dimov I. (2015) An introduction to applied quantum mechanics in the Wigner Monte Carlo formalism, Physics Reports Volume 577: 1–34.

[11] Shao S., Lu T., Cai W. (2011) Adaptive conservative cell average spectral element methods for transient Wigner equation in quantum transport. Commun. Comput. Phys., 9: 711–739.

[12] Shao S. and Sellier J.M. (2015) Comparison of deterministic and stochastic methods for time-dependent Wigner simulations. J. Comput. Phys., 300: 167–185.

[13] Todorov, V., Dimov, I., Georgieva, R., & Dimitrov, S. (2019). Adaptive Monte Carlo algorithm for Wigner kernel evaluation. Neural Computing and Applications, 1-12.

[14] Xiong Y., Chen Z., Shao S (2016) An advective-spectral-mixed method for time-dependent many-body Wigner simulations. SIAM J. Sci. Comput., to appear, [arXiv:1602.08853].

[15] Wigner E. (1932) On the quantum correction for thermodynamic equilibrium, Phys. Rev. 40: 749.

Proceedings of the 16th Conference on Computer
Science and Intelligence Systems pp. 239–242

DOI: 10.15439/2021F54
ISSN 2300-5963 ACSIS, Vol. 25

Optimized stochastic approach for integral equations

Venelin Todorov
Bulgarian Academy of Sciences
Institute of Mathematics and Informatics
ul. G. Bonchev 8, 1113 Sofia, Bulgaria
Bulgarian Academy of Sciences
Institute of Information and Communication Technologies
ul. G. Bonchev 25A, 1113 Sofia, Bulgaria
Email: vtodorov@math.bas.bg,venelin@parallel.bas.bg

Stefka Fidanova
Bulgarian Academy of Sciences
Institute of Information and Communication Technologies
ul. G. Bonchev 25A, 1113 Sofia, Bulgria
Email: stefka@parallel.bas.bg

Ivan Dimov
Bulgarian Academy of Sciences
Institute of Information and Communication Technologies
ul. G. Bonchev 25A, 1113 Sofia, Bulgria
Email: ivdimov@bas.bg

Rayna Georgieva
Bulgarian Academy of Sciences
Institute of Information and Communication Technologies
ul. G. Bonchev 25A, 1113 Sofia, Bulgria
Email: rayna@parallel.bas.bg

***Abstract*—An optimized Monte Carlo approach (OPTIMIZED MC) for a Fredholm integral equations of the second kind is presented and discussed in the present paper. Numerical examples and results are discussed and MC algorithms with various initial and transition probabilities are compared.**

I. Introduction

INTEGRAL equations are of high importancy in various areas of applied mathematics [7]. That is why it is important to construct effective methods to solve integral equations. An important advantage of Monte Carlo (MC) methods is that they allow to search an unknown linear functional of the solution directly [1].

II. Formulation of the problem

The Fredholm integral equation of the second kind has been analyzed:

$$u(x) = \int_\Omega k(x,x')u(x')\,dx' + f(x) \ \ or \ \ u = \mathcal{K}u + f, \quad (1)$$

where

$$x, x' \in \Omega \subset \mathbb{R}^d, \ \ u(x), f(x) \in L_2(\Omega), \ \ k(x,x') \in L_2(\Omega \times \Omega)$$

and $\mathcal{K}$ is the integral operator. Usually a linear functional from the solution:

$$J(u) = \int \varphi(x)u(x)dx = (\varphi, u) \quad (2)$$

should be evaluated in various problems. A MC algorithm is described below. Let $\varphi(x) \in L_2(\Omega)$. A set of permissible densities is defined:

$$\pi(x), \ p(x,x') : \ \pi(x) \ge 0, \ p(x,x') \ge 0,$$

$$\int_\Omega \pi(x)\,dx = 1, \int_\Omega p(x,x')\,dx' = 1, \ x \in \Omega \subset \mathbb{R}^d.$$

We define a Markov chain $T_k \ : \ x_0 \to x_1 \to \cdots \to x_k$ [4] with length k started from the initial state x_0. If the approximate initial solution coincides with the corresponding right-hand side $f(x)$, a MC algorithm for integral equations [6] is defined by:

$$E\theta_k[\varphi] = \left(\varphi, u^{(k)}\right), \ \theta_k[\varphi] = \frac{\varphi(x_0)}{\pi(x_0)} \sum_{j=0}^{k} W_j f(x_j),$$

$$W_0 = 1, \ W_j = W_{j-1}\frac{k(x_{j-1},x_j)}{p(x_{j-1},x_j)}, \ j = 1,\ldots,k,$$

$$\left(\varphi, u^{(k)}\right) \approx \frac{1}{N}\sum_{n=1}^{N} \theta_k[\varphi]_n.$$

III. A probabilistic error estimate

The probabilistic error is $r_N \le 0.6745\sigma(\theta)\frac{1}{\sqrt{N}}$ [3], [2], where N is the number of samples of the random variable θ and $\sigma(\theta) = (D\theta)^{1/2}$ is the standard deviation of the random variable θ for which $E\theta_k[\varphi] = \left(\varphi, u^{(k)}\right) = \sum_{j=0}^{k}(\varphi, \mathcal{K}^{(j)}f)$, where for point $x = (x_0, \ldots, x_j) \in G \equiv \Omega^{j+1} \subset \mathbb{R}^{d(j+1)}$, $j = 1, \ldots, k$:

$$(\varphi, \mathcal{K}^{(j)}f) = \int_\Omega \varphi(x_0)\mathcal{K}^{(j)}f(x_0)dx_0 =$$

$$= \int_G \varphi(x_0)k(x_0,x_1)\ldots k(x_{k-1},x_j)f(x_j)dx_0dx_1\ldots dx_j =$$

The work is supported by KP-06-M32/2-17.12.2019 'Advanced Stochastic and Deterministic Approaches for Large-Scale Problems of Computational Mathematics" and by the National Scientific Program "Information and Communication Technologies for a Single Digital Market in Science, Education and Security" (ICTinSES), contract No. D01-205/23.11.2018, financed by the Ministry of Education and Science. The work is supported by the Bulgarian National Science Fund under Project DN 12/5-2017 "Efficient Stochastic Methods and Algorithms for Large-Scale Problems" and by the Project KP-06-Russia/17 "New Highly Efficient Stochastic Simulation Methods and Applications" funded by the National Science Fund - Bulgaria.

IEEE Catalog Number: CFP2185N-ART ©2021, PTI

$$\int_G F(x)dx,$$

where

$$F(x) = \varphi(x_0)k(x_0,x_1)\dots k(x_{k-1},x_j)f(x_j),\ x \in G \subset \mathbb{R}^{d(j+1)}.$$

Using the inequality $D\sum\limits_{j=0}^{k}\theta_k^{(j)} \le \left(\sum\limits_{j=0}^{k}\sqrt{D\theta_k^{(j)}}\right)^2$, and the variance properties we have the following inequalities [1]:

$$r_N \le$$

$$\frac{0.6745}{\sqrt{N}}\sum_{j=0}^{k}\left(\int_G\left(\mathcal{K}^{(j)}\varphi f\right)^2 pdx - \left(\int_G \mathcal{K}^{(j)}\varphi f pdx\right)^2\right)^{1/2} \le$$

$$\le \frac{0.6745}{\sqrt{N}}\sum_{j=0}^{k}\left(\int_G\left(\mathcal{K}^{(j)}\varphi f\right)^2 pdx\right)^{1/2}$$

$$= \frac{0.6745}{\sqrt{N}}\|\varphi\|_{L_2}\|f\|_{L_2}\sum_{j=0}^{k}\left\|\mathcal{K}^{(j)}\right\|_{L_2}.$$

The following estimate is obtained:

$$r_N \le \frac{0.6745\|f\|_{L_2}\|\varphi\|_{L_2}}{\sqrt{N}\left(1-\|\mathcal{K}\|_{L_2}\right)}.$$

IV. A Systematic Error Estimate

The sequence [2] $u^{(1)},\ u^{(2)},\dots$ is defined by the recursion formula $u^{(k)} = \mathcal{K}u^{(k-1)}+f, k = 1,2,\dots$. The formal solution of the equation (1) is the truncated Neumann series $u^{(k)} = f + \mathcal{K}f + \dots + \mathcal{K}^{(k-1)}f + \mathcal{K}^{(k)}u^{(0)}, k > 0$, where the k^{th} iteration of $\mathcal{K}$ is denoted by $\mathcal{K}^{(k)}$, and $u^{(k)} = \sum\limits_{i=0}^{k-1}\mathcal{K}^{(i)}f + \mathcal{K}^{(k)}u^{(0)}$.

We construct the k - residual vector of the systematic error $r^{(k)}$: $r^{(k)} = f - (I-\mathcal{K})\,u^{(k)} = (I-\mathcal{K})\left(u-u^{(k)}\right)$.

By the definition of $r^{(k)}$: $r^{(k)} = f - u^{(k)} + \mathcal{K}u^{(k)} = u^{(k+1)} - u^{(k)}$ and $r^{(k+1)} = u^{(k+2)} - u^{(k+1)} = \mathcal{K}u^{(k+1)} + f - \mathcal{K}u^{(k)} - f = \mathcal{K}\left(u^{(k+1)} - u^{(k)}\right) = \mathcal{K}r^{(k)}$.

We have $r^{(0)} = u^{(1)} - u^{(0)} = \mathcal{K}u^{(0)} + f - u^{(0)} = \mathcal{K}f$, $r^{(k+1)} = \mathcal{K}r^{(k)} = \mathcal{K}^{(2)}r^{(k-1)} = \dots = \mathcal{K}^{(k+1)}r^{(0)}$.

So we obtain $u^{(k+1)} = u^{(k)} + r^{(k)} = u^{(k-1)} + r^{(k-1)} + r^{(k)} = \dots = u^{(0)} + r^{(0)} + \dots + r^{(k)} = u^{(0)} + r^{(0)} + \mathcal{K}r^{(0)} + \mathcal{K}^{(2)}r^{(0)} + \dots + \mathcal{K}^{(k)}r^{(0)} = u^{(0)} + \left(I + \mathcal{K} + \dots + \mathcal{K}^{(k)}\right)r^{(0)}$ [4].

If $\|\mathcal{K}\|_{L_2} < 1$ then the Neumann series $u = \sum\limits_{i=0}^{\infty}\mathcal{K}^{(i)}f$ is convergent and $u^{(k+1)} \xrightarrow{k\to\infty} u$ therefore from $u^{(k+1)} = u^{(0)} + \left(I + \mathcal{K} + \dots + \mathcal{K}^{(k)}\right)r^{(0)}$ and $k \to \infty$ we have $u = u^{(0)} + (I-\mathcal{K})^{-1}r^{(0)}$. After simple transformations $u = \mathcal{K}u + f = \mathcal{K}u^{(0)} + \mathcal{K}(I-\mathcal{K})^{-1}r^{(0)} + f = u^{(1)} + \mathcal{K}(I-\mathcal{K})^{-1}r^{(0)}$.

Doing this k times we obtain: $u = u^{(k)} + \mathcal{K}^{(k)}(I-\mathcal{K})^{-1}r^{(0)}$. The following inequalities are established applying the Cauchy-Schwarz inequality:

$$r^{(k)} = \left\|u - u^{(k)}\right\|_{L_2} \le$$

$$\frac{\|\mathcal{K}\|_{L_2}^{k}\left\|r^{(0)}\right\|_{L_2}}{1-\|\mathcal{K}\|_{L_2}} \le \frac{\|\mathcal{K}\|_{L_2}^{k}\|f\|_{L_2}\|\mathcal{K}\|_{L_2}}{1-\|\mathcal{K}\|_{L_2}} =$$

$$\frac{\|\mathcal{K}\|_{L_2}^{k+1}\|f\|_{L_2}}{1-\|\mathcal{K}\|_{L_2}}.$$

The systematic error is estimated in following way:

$$\left|(\varphi,u) - \left(\varphi,u^{(k)}\right)\right| \le \|\varphi\|_{L_2}\left\|u-u^{(k)}\right\|_{L_2} \le$$

$$\frac{\|\varphi\|_{L_2}\|f\|_{L_2}\|\mathcal{K}\|_{L_2}^{k+1}}{1-\|\mathcal{K}\|_{L_2}}.$$

V. The Optimized Stochastic Approach

Let us denote by δ an accuracy to solve the task under consideration (2). This means that:

$$r_N \le \frac{0.6745\|\varphi\|_{L_2}\|f\|_{L_2}}{\sqrt{N}\left(1-\|\mathcal{K}\|_{L_2}\right)} \le \frac{\delta}{2},$$

$$r_k \le \frac{\|\varphi\|_{L_2}\|f\|_{L_2}\|\mathcal{K}\|_{L_2}^{k+1}}{1-\|\mathcal{K}\|_{L_2}} \le \frac{\delta}{2}.$$

For a Fredholm integral equation (1) the lower bounds for N and k for the OPTIMIZED MC algorithm are:

$$N \ge \left(\frac{1.349\|\varphi\|_{L_2}\|f\|_{L_2}}{\delta\left(1-\|\mathcal{K}\|_{L_2}\right)}\right)^2,\ k \ge \frac{\ln\frac{\delta\left(1-\|\mathcal{K}\|_{L_2}\right)}{2\|\varphi\|_{L_2}\|f\|_{L_2}\|\mathcal{K}\|_{L_2}}}{\ln\|\mathcal{K}\|_{L_2}}.$$

We have also obtained an optimal ratio between k and N: For a Fredholm integral equation (1) the lower bounds for N and k for the OPTIMIZED MC algorithm are:

$$N \ge \left(\frac{1.349\|\varphi\|_{L_2}\|f\|_{L_2}}{\delta\left(1-\|\mathcal{K}\|_{L_2}\right)}\right)^2,\ k \ge \frac{\ln\frac{0.6745}{\|\mathcal{K}\|_{L_2}\sqrt{N}}}{\ln\|\mathcal{K}\|_{L_2}}.$$

VI. Numerical Examples and Results

A. Example 1

The first example is:

$$u\left(x\right) = \int_{\Omega} k\left(x,x'\right)u\left(x'\right)dx' + f\left(x\right),$$

$\Omega \equiv [0,1]$, $k\left(x,x'\right) = \frac{1}{6}e^{x+x'}$, $f\left(x\right) = 6x - e^x$. $\varphi(x)$ is the delta function $(\Delta(x))$. The exact solution is $u(x) = 6x$. We are interested in the value of the solution at the middle of the interval. Firstly, the L_2 norms are computed: $\|\varphi\|_{L_2} = 1$, $\|\mathcal{K}\|_{L_2} = 0.5324$, $\|f\|_{L_2} = 1.7873$. For this example the exact solution is 3 and $\pi\left(x\right) = \Delta(x)$. We make 20 algorithm runs on Intel Core i5-2410M @ 2.3 GHz.

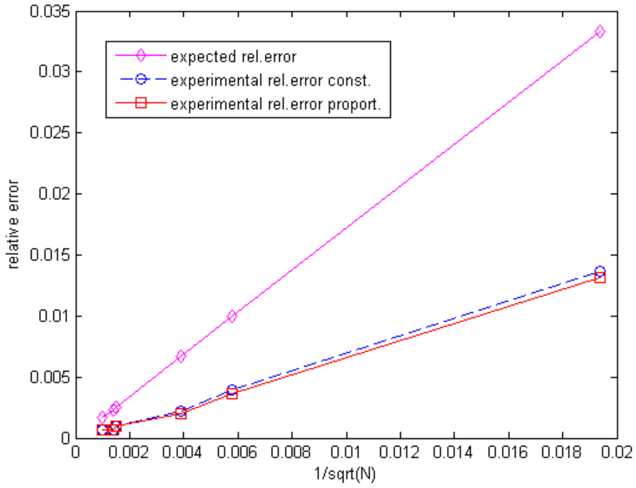
TABLE I
RESULTS FOR THE FIRST EXAMPLE.

δ	N	k	expected	BASIC	time	OPTIMIZED	time
			rel. error	rel. error	(sec.)	rel. error	(sec.)
0.1	2659	6	0.0333	0.0137	11	0.0132	5
0.03	29542	8	0.01	0.0039	62	0.0036	42
0.02	66468	9	0.0067	0.0022	140	0.0020	70
0.0075	472659	10	0.0025	0.001	1167	9.3671e-04	529
0.007	542593	11	0.00233	6.9639e-04	1562	6.3582e-04	614
0.005	1063482	11	0.00167	6.4221e-04	4412	6.2479e-04	2202

TABLE II
RESULTS FOR THE SECOND EXAMPLE.

δ	N	k	expected	BASIC	time	OPTIMIZED	time
			rel. error	rel. error	sec.	rel. error	sec.
0.23	132	3	0.1395	0.0123	0.5	0.0121	0.2
0.037	5101	4	0.0224	0.0041	11	0.0040	7
0.025	11172	5	0.0152	0.0014	16	0.0012	9
0.014	35623	6	0.0085	4.5725e-04	56	4.0010e-04	34
0.0055	230809	7	0.0033	1.5242e-04	424	9.8811e-05	346
0.0045	344788	7	0.0027	1.5242e-04	605	1.4893e-04	592

Fig. 1. Experimental and expected relative error.



B. Numerical results for the first example

The first two columns with the expected relative error (RE) and the computational time (CPU time) measured in seconds are for the case when the transition probabilities are constant functions (this is the standard MC method and we use the notation BASIC) and the last two columns are for the case when OPTIMIZED is used (this is also called the almost optimal MC algorithm). From the Tables it leads that the OPTIMIZED method gives better results (smaller relative errors and significantly smaller computational times).

We can see the comparison between the expected and experimental relative error on Figure 1 which shows that experimental RE confirms the expected RE.

The OPTIMIZED MC algorithm has a higher computational efficiency than the BASIC MC algorithm because its CPU time is smaller.

C. Example 2

The next example is a biology analytically tractable model [5]:

$$u\left(x\right) = \int_{\Omega} k\left(x, x'\right) u\left(x'\right) dx' + f\left(x\right),$$

$\Omega \equiv [0,1]$, $k\left(x, x'\right) = \frac{1}{3}e^{x}$, $f\left(x\right) = \frac{2}{3}e^{x}$. $\varphi(x)$ is the delta function. The exact solution is $u(x) = e^{x}$. We are interested in the value of the solution at the middle point of the interval. The L_2 norms are evaluated as follows: $\|\varphi\|_{L_2} = 1$, $\|\mathcal{K}\|_{L_2} = 0.3917$, $\|f\|_{L_2} = 1.1915$. Here the exact solution is 1.6487 and $\pi\left(x\right) = \Delta(x)$. We make 20 algorithm runs on the same computational unit.

D. Numerical results for the second example

One can see that the OPTIMIZED method gives slightly better results than the BASIC MC and the results are closer when the initial probability is the delta function. However again the OPTIMIZED MC algorithm has a higher computational efficiency than the BASIC algorithm because its CPU time is shorter.

E. Example 3

We study the following example describes the procedure of teaching of neural networks [4], [5]:

$$u\left(x\right) = \int_{\Omega} k\left(x, x'\right) u\left(x'\right) dx' + f\left(x\right),$$

$\Omega \equiv [-2,2]$, $k\left(x, x'\right) = \frac{0.055}{1+e^{-3x}} + 0.07$, $f\left(x\right) = 0.02\left(3x^2 + e^{-0.35x}\right)$, $\varphi(x) = 0.7((x+1)^2 \cos(5x) + 20)$.

Here $\varphi(x) = 0.7((x+1)^2 \cos 5x + 20)$. The exact solution is 8.98635750518 [2]. We calculate: $\|\varphi\|_{L_2} = 27.7782$, $\|\mathcal{K}\|_{L_2} = 0.2001$, $\|f\|_{L_2} = 0.2510$.
We make 20 algorithm runs on the same processor.

F. Numerical results for the third example

The results presented in Table III demonstrates that the OPTIMIZED MC method gives much smaller relative errors than the BASIC MC algorithm for larger values of N and k. In the case of smaller values of these quantities the BASIC MC gives smaller relative errors, but the RE obtained with OPTIMIZED method are closer to the expected RE. Using the OPTIMIZED approach we see that the experimental RE confirms the expected RE. We also see that in the OPTIMIZED algorithm is a little bit slower because we use the acceptance-rejection method for modeling the initial probabilities.

TABLE III
RESULTS FOR THE THIRD EXAMPLE.

δ	N	k	expected	BASIC	time	OPTIMIZED	time
			rel. error	rel. error	sec.	rel. error	sec.
0.4	865	3	0.0445	0.0052	3	0.0239	5
0.2	3457	4	0.0223	0.0094	9	0.0121	23
0.1	13827	4	0.0111	0.0113	28	0.0086	46
0.05	55306	5	0.00556	0.0177	132	0.0032	222
0.028	176357	5	0.00312	0.0176	448	0.0031	540
0.02	345659	6	0.00233	0.0202	901	0.0013	1090

VII. CONCLUSION

In this paper we present an optimized stochastic algorithm for solving the Fredholm integral equation of the second kind. Two main cases are taken into account - the initial probability coincides with the delta function, and the second case when the initial probability is different from the delta function. The results from the numerical tests in the first case show that the OPTIMIZED MC reaches much smaller computational times than the BASIC MC with constant probabilities and comparable relative errors respectively. The results from the numerical tests in the second case show that the OPTIMIZED MC reaches much smaller relative errors than the BASIC MC with constant probabilities and comparable computational times respectively. the OPTIMIZED MC has a higher computational efficiency than the BASIC MC. The main conclusion here is that the OPTIMIZED MC approach is characterized by a higher computational efficiency (proportional to relative error and compuatational time) in both cases under consideration.

REFERENCES

[1] I. Dimov, Monte Carlo Methods for Applied Scientists, New Jersey, London, Singapore, World Scientific, 2008, 291p.

[2] R. Georgieva, PhD Thesis: Computational complexity of Monte Carlo algorithms for multidimensional integrals and integral equations, Sofia, 2003

[3] I. Dimov, E. Atanassov, What Monte Carlo models can do and cannot do efficiently?, *Applied Mathematical Modelling* **32** (2007) 1477–1500.

[4] J.H. Curtiss. Monte Carlo Methods for The Iteration of Linear Operators. *J. Math. Phys.*, **32** 209–232, (1954).

[5] A. Doucet, A.M. Johansen, V.B. Tadic. On solving integral equations using Markov chain Monte Carlo methods. *Applied Mathematics and Computations*, **216** 2869–2880, (2010).

[6] I. Sobol. Numerical methods Monte Carlo. Nauka, Moscow, 1973.

[7] S. L. Zaharieva, I. Radoslavov Georgiev, V. A. Mutkov and Y. Branimirov Neikov, "Arima Approach For Forecasting Temperature In A Residential Premises Part 2," 2021 20th International Symposium Infoteh-jahorina (infoteh), 2021, pp. 1-5.

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 243–246

DOI: 10.15439/2021F53
ISSN 2300-5963 ACSIS, Vol. 25

Optimized Method based on Lattice Sequences for Multidimensional Integrals in Neural Networks

Venelin Todorov *†, Ivan Dimov†, Stefka Fidanova †
*Institute of Mathematics and Informatics
Bulgarian Academy of Sciences
8 Acad. G. Bonchev Str., 1113 Sofia, Bulgaria
†Institute of Information and Communication Technologies
Bulgarian Academy of Sciences
25A Acad. G. Bonchev Str., 1113 Sofia, Bulgaria
Email: vtodorov@math.bas.bg, venelin@parallel.bas.bg, ivdimov@bas.bg, stefka@parallel.bas.bg

***Abstract*—In this work we investigate advanced stochastic methods for solving a specific multidimensional problem related to neural networks. Monte Carlo and quasi-Monte Carlo techniques have been developed over many years in a range of different fields, but have only recently been applied to the problems in neural networks. As well as providing a consistent framework for statistical pattern recognition, the stochastic approach offers a number of practical advantages including a solution to the problem for higher dimensions. For the first time multidimensional integrals up to 100 dimensions related to this area will be discussed in our numerical study.**

I. Introduction

IN 2011 Shaowei Lin in his works [5],[6] consider the problem of evaluating multidimensional integrals in Bayesian statistics which are used in neural networks. The first has the form

$$\int_{\Omega} p_1^{u_1}(x) \dots p_s^{u_s}(x) dx, \qquad (1)$$

where $\Omega \in \mathcal{R}^s,\ x = (x_1, \dots, x_s)$, $p_i(x)$ are polynomials and u_i are integers. The second kind of integrals has the form

$$\int_{\Omega} e^{-Nf(x)} \phi(x) dx, \qquad (2)$$

where $f(x)$ and $\phi(x)$ are multidimensional polynomials and N is an integer number. These integrals are evaluated unsatisfactory with deterministic [11] and algebraic methods [9] up to now, and it is known that Monte Carlo methods [3] outperform these methods especially for high dimensions [12].

We will now give a brief explanation which demonstrates the strength of the MC and QMC approach [3]. According to [3] we will choose 100 nodes on the each of the coordinate axes in the s-dimensional cube $G = E^s$ and we have to evaluate about 10^{100} values of the function $f(x)$. Assume a time of $10^{-7}s$ is necessary for calculating one value of the function [3]. So, a time of order 10^{93}s will be necessary for computation of the integral, and 1 year has 31536×10^3s.

Now MC approach consists of generating N pseudo random values (points) (PRV) in G; in evaluating the values of $f(x)$ at these points; and averaging the computed values of the function. For each uniformly distributed random (UDR) point in G we have to generate 100 UDR numbers in $[0, 1]$. Assume that the expression in front of h^{-6} is of order 1 [3]. Here $h = 0.1$, and we have $N \approx 10^6$; so, it will be necessary to generate $100 \times 10^6 = 10 \times 10^7$ PRV. Usually, 2 operations are sufficient to generate a single PRV. According to [3] the time required to generate one PRV is the same as that for computation the value of $f(x)$. So, in order to solve the task with the same accuracy, a time of

$$10 \times 10^7 \times 2 \times 10^{-7} \approx 20s$$

will be necessary. We summarize that in the case of 100-dimensional integral it is 5×10^{91} times faster than the deterministic one. That motivates our study on the new highly efficient stochastic approaches for the problem under consideration.

II. The new stochastic approach

We will use this rank-1 lattice sequence [10]:

$$\mathbf{x}_k = \left\{ \frac{k}{N} \mathbf{z} \right\},\ k = 1, \dots, N, \qquad (3)$$

where N is an integer, $N \geq 2$, $\mathbf{z} = (z_1, z_2, \dots z_s)$ is the generating vector and $\{z\}$ denotes the fractional part of z. For the definition of the $E_s^{\alpha}(c)$ and $P_{\alpha}(z, N)$ see [10] and for more details, see also [1].

Definition 1: Consider the point set $X = \{x_i \mid i = 1, 2, \dots N\}$ in $[0, 1)^s$ and $N > 1$. Denote by $x_i = (x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(s)})$ and $J(v) = [0, v_1) \times [0, v_2) \times \dots \times [0, v_s)$. Then the discrepancy of the set is defined as

$$D_N^* := \sup_{0 \leq v_j \leq 1} \left| \frac{\#\{x_i \in J(v)\}}{N} - \prod_{j=1}^{s} v_j \right|. \qquad (4)$$

Venelin Todorov is supported by the Bulgarian National Science Fund under Project KP-06-M32/2 - 17.12.2019 ”Advanced Stochastic and Deterministic Approaches for Large-Scale Problems of Computational Mathematics” and by the National Scientific Program ”Information and Communication Technologies for a Single Digital Market in Science, Education and Security (ICT in SES)”, contract No DO1-205/23.11.2018, financed by the Ministry of Education and Science in Bulgaria. The work is also supported by Bulgarian National Science Fund under Project DN 12/5-2017 ”Efficient Stochastic Methods and Algorithms for Large-Scale Problems” and by the Project KP-06-Russia/17 ”New Highly Efficient Stochastic Simulation Methods and Applications” funded by the National Science Fund - Bulgaria.

In 1959 Bahvalov proved that [1] there exists an optimal choice of the generating vector $\mathbf{z}$:

$$\left| \frac{1}{N} \sum_{k=1}^{N} f\left(\left\{ \frac{k}{N}\mathbf{z} \right\} \right) - \int\limits_{[0,1)^s} f(u)du \right| \leq cd(s,\alpha) \frac{(\log N)^{\beta(s,\alpha)}}{N^{\alpha}}, \tag{5}$$

for the function $f \in E_s^{\alpha}(c)$, $\alpha > 1$ and $d(s,\alpha), \beta(s,\alpha)$ does not depend on N.

The generating vector $\mathbf{z}$ which satisfies (5), is an optimal generating vector [10] and while the existence of optimal generating vectors is proved by the theoretical result, the main bottleneck lies in the construction of the optimal vectors, especially for very high dimensions [3].

The first generating vector in our study is the generalized Fibonacci numbers of the corresponding dimension:

$$\mathbf{z} = (1, F_n^{(s)}(2), \ldots, F_n^{(s)}(s)), \tag{6}$$

where we use that $F_n^{(s)}(j) := F_{n+j-1}^{(s)} - \sum\limits_{i=0}^{j-2} F_{n+i}^{(s)}$ and $F_{n+l}^{(s)}$ $(l = 0, \ldots, j-1, j$ is an integer, $2 \leq j \leq s)$ is the term of the s-dimensional Fibonacci sequence [10].

If we change the generating vector to be optimal in the way described in [4] we have improved the lattice sequence. We will now give the description of the steps of our algorithms. At the beginning of the algorithm the input is the number of dimensionality s and the number of samples N. At the first step of the algorithm s dimensional optimal generating vector

$$\mathbf{z} = (z_1, z_2, \ldots z_s) \tag{7}$$

is generated by the fast construction method described by Dirk Nuyens [4]. The second step of the algorithm includes generating the points of lattice rule by formula

$$\mathbf{x}_k = \left\{ \frac{k}{N}\mathbf{z} \right\}, \ k = 1, \ldots, N. \tag{8}$$

And at the third and last step of the algorithm an approximate value I_N of the multidimensional integral is evaluated by the formula:

$$I_N = \frac{1}{N} \sum_{k=1}^{N} f\left(\left\{ \frac{k}{N}\mathbf{z} \right\} \right). \tag{9}$$

The special choice of this optimal generating vector is definitely more efficient than the Fibonacci generating vector, which is only optimal for the two dimensional case [10]. For our improved lattice rule is satisfied [4]:

$$D_N^* = \mathcal{O}\left(\frac{log^s N}{N} \right). \tag{10}$$

The steps of working of the algorithm are given on the flowchart on Fig. 1.

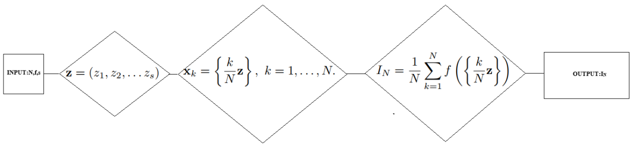

INPUT:N,f,s
$\mathbf{z} = (z_1, z_2, \ldots z_s)$
$\mathbf{x}_k = \left\{ \frac{k}{N}\mathbf{z} \right\}, \ k = 1, \ldots, N.$
$I_N = \frac{1}{N} \sum_{k=1}^{N} f\left(\left\{ \frac{k}{N}\mathbf{z} \right\} \right)$
OUTPUT:I_N

Figure 1. The flowchart of the optimized lattice algorithms

III. Numerical results

We considered different examples of 4,7,10,30 and 100 dimensional integrals, respectively, for which we have computed their referent values.

Example 1. s = 4.

$$\int\limits_{[0,1]^4} x_1 x_2^2 e^{x_1 x_2} \sin(x_3) \cos(x_4) \approx 0.108975. \tag{11}$$

Example 2. s = 7.

$$\int\limits_{[0,1]^7} e^{1 - \sum\limits_{i=1}^{3} sin(\frac{\pi}{2}.x_i)} .arcsin \left(sin(1) + \frac{\sum\limits_{j=1}^{7} x_j}{200} \right) \approx 0.7515. \tag{12}$$

Example 3. s = 10.

$$\int\limits_{[0,1]^{10}} \frac{4x_1 x_3^2 e^{2x_1 x_3}}{(1 + x_2 + x_4)^2} e^{x_5 + \cdots + x_{10}} \approx 14.808435. \tag{13}$$

Example 4. s= 30.

$$\int\limits_{[0,1]^{30}} \frac{4x_1 x_3^2 e^{2x_1 x_3}}{(1 + x_2 + x_4)^2} e^{x_5 + \cdots + x_{20}} x_{21} \ldots x_{30} \approx 3.244. \tag{14}$$

We also consider the 100-dimensional multidimensional integral defined by the following way:

Example 5. s= 100.

$$I_{100} = \int\limits_{[0,1]^{100}} \exp\left(\prod_{i=1}^{100} x_i \right), \tag{15}$$

whose reference value is calculated by expanding the exponential function in Taylor series and integrating the terms $(x_1 \cdots x_{100})^n$ namely

$$\int_{[0,1]^{100}} \exp\left(\prod_{i=1}^{100} x_i \right) =$$

$$= \sum_{n=0}^{\infty} \frac{1}{(n+1)^{100} n!} =_{100} F_{100}(1, \cdots, 1; 2, \cdots, 2; 1),$$

where ${}_pF_q(a_1, \cdots, a_p; b_1, \cdots, b_q; x)$ is the generalized hypergeometric function

$${}_pF_q(a_1, \cdots, a_p; b_1, \cdots, b_q; x) = \sum_{n=0}^{\infty} \frac{(a_1)_n \cdots (a_p)_n}{(b_1)_n \cdots (b_q)_n} \frac{x^n}{n!},$$

and $(c)_n = c(c+1) \cdots (c+n-1)$ is the Pochhammer symbol.

We make a comparison between the optimized lattice sequence with an optimal generating vector (OPT), Fibonacci lattice sets (FIBO), Latin hypercube sampling (LHS) [7] and the scrambled Sobol sequence (SOBOLS) [8]. Each Table below contains information about the stochastic approach which is applied, the obtained relative errors (REs), the needed CPU-time in seconds and the number of points. Note that when the FIBO method is tested, the number of sampled points are always generalized Fibonacci numbers of the corresponding dimensionality. The computer working architecture is Core i7-4710MQ at 2.50GHz and 8GB of RAM. We performs 10 algorithmic runs using MATLAB on CPU Core i7-4710MQ for the algorithms to validate our assumptions of experimentation.

Table I
ALGORITHM COMPARISON OF THE REs FOR THE 4-DIMENSIONAL INTEGRAL FOR DIFFERENT NUMBER OF POINTS.

# of points	OPT	t,s	FIBO	t,s	LHS	t,s	SOBOLS	t,s
1490	6.11e-4	0.002	1.01e-3	0.004	8.16e-4	0.005	3.78e-3	0.47
10671	2.13e-5	0.01	8.59e-5	0.02	6.11e-4	0.01	6.10e-4	1.59
20569	6.56e-6	0.02	3.89e-5	0.03	5.01e-5	0.02	1.97e-5	4.54
39648	9.14e-7	0.06	3.01e-5	0.07	4.18e-5	7.09	9.67e-6	8.26
147312	4.78e-7	0.15	3.71e-6	0.24	2.19e-5	0.28	1.40e-6	27.91

Table II
ALGORITHM COMPARISON OF THE REs FOR THE 4-DIMENSIONAL INTEGRAL FOR A PRELIMINARY GIVEN TIME.

t, s	OPT	FIBO	LHS	SOBOLS
1	5.66e-7	5.62e-6	1.54e-5	6.32e-4
5	3.12e-7	5.38e-7	9.18e-6	1.23e-5
10	5.14e-8	3.77e-7	6.51e-6	8.48e-6
20	3.18e-8	2.67e-8	2.31e-6	1.16e-6

Table III
ALGORITHM COMPARISON OF THE REs FOR THE 7-DIMENSIONAL INTEGRAL FOR DIFFERENT NUMBER OF POINTS.

# of points	OPT	t,s	FIBO	t,s	LHS	t,s	SOBOLS	t,s
2000	6.39e-4	0.14	2.81e-3	0.23	5.45e-3	0.25	2.51e-3	1.42
7936	3.23e-4	0.64	1.38e-3	0.87	2.11e-3	0.91	1.16e-3	3.08
15808	1.23e-5	0.95	9.19e-4	1.73	8.31e-4	1.81	7.58e-4	5.89
62725	3.15e-6	2.54	2.78e-5	3.41	6.22e-4	3.5	3.11e-4	15.64
124946	1.12e-6	6.48	6.87e-5	6.90	4.34e-4	7.1	8.22e-5	31.41

Table IV
ALGORITHM COMPARISON OF THE REs FOR THE 7-DIMENSIONAL INTEGRAL FOR A PRELIMINARY GIVEN TIME.

t, s	OPT	FIBO	LHS	SOBOLS
0.1	7.38e-4	2.38e-3	6.65e-3	8.37e-3
1	1.17e-5	6.19e-4	3.05e-3	1.37e-3
5	2.32e-6	8.81e-5	4.89e-4	8.38e-4
10	9.11e-7	1.88e-5	2.16e-4	4.78e-4
20	7.43e-7	3.87e-6	8.56e-5	9.87e-5

Table V
ALGORITHM COMPARISON OF THE REs FOR THE 10-DIMENSIONAL INTEGRAL FOR DIFFERENT NUMBER OF POINTS.

# of points	OPT	t,s	FIBO	t,s	LHS	t,s	SOBOLS	t,s
1597	3.14e-4	0.002	4.39e-3	0.003	7.31e-3	0.01	1.46e-3	0.05
17711	6.21e-5	0.02	1.81e-3	0.04	4.45e-3	0.07	1.83e-4	0.21
121393	4.34e-6	0.15	1.20e-3	0.16	7.23e-4	0.21	3.12e-5	1.47
832040	4.11e-7	0.75	1.19e-5	0.70	3.11e-4	0.83	8.25e-6	14.41
3524578	5.32e-8	6.35	2.63e-6	6.45	8.57e-5	6.7	7.71e-7	139.1

Table VI
ALGORITHM COMPARISON OF THE REs FOR THE 10-DIMENSIONAL INTEGRAL FOR A PRELIMINARY GIVEN TIME.

t, s	OPT	FIBO	LHS	SOBOLS
0.1	4.95e-6	9.19e-6	4.13e-3	4.19e-4
1	8.10e-7	5.63e-6	2.55e-4	1.21e-4
5	3.56e-8	2.15e-6	1.23e-4	7.21e-5
10	4.31e-8	1.79e-6	7.17e-5	3.51e-5
20	9.13e-9	8.61e-7	3.42e-5	7.09e-6

Table VII
ALGORITHM COMPARISON OF THE REs FOR THE 30-DIMENSIONAL INTEGRAL FOR DIFFERENT NUMBER OF POINTS.

# of points	OPT	t,s	SOBOLS	t,s	LHS	t,s	FIBO	t,s
1024	1.21e-2	0.02	5.78e-2	0.53	5.68e-2	0.03	8.81e-1	0.02
16384	4.11e-3	0.16	1.53e-2	5.69	8.60e-3	0.18	6.19e-1	0.14
131072	5.24e-4	1.34	1.35e-3	42.1	5.38e-3	1.2	2.78e-1	1.16
1048576	8.81e-5	9.02	6.78e-4	243.9	9.31e-4	8.9	9.86e-2	8.61

Table VIII
ALGORITHM COMPARISON OF THE REs FOR THE 30-DIMENSIONAL INTEGRAL FOR A PRELIMINARY GIVEN TIME.

t, s	OPT	SOBOLS	LHS	FIBO
1	3.48e-3	2.38e-2	7.21e-3	2.38e-1
5	4.23e-4	5.46e-3	5.16e-3	1.81e-1
10	8.91e-5	1.25e-3	8.21e-4	9.48e-2
20	2.33e-5	6.11e-4	4.35e-4	7.87e-2

Numerical results show significant advantage for the optimized lattice sets algorithm based on an optimal generating vector in comparison with FIBO, LHS and SOBOLS scramble sequence (1-2 orders). For the 4-th dimensional integral the best approach is produced by the optimized method OPT - a relative error $4.78e-7$ for $N = 147312$ - see Table I and for 20s the best approach is FIBO - $2.67e-8$ in Table II with two orders better results than both SOBOLS and LHS. For the 7-th dimensional integral the best approach is produced by the optimized method OPT - a relative error $1.12e-6$ for $N = 124946$ - see Table III and for 20s the best approach is OPT - $7.43e-7$ in Table IV with one order better REs than FIBO and two order better REs than both SOBOLS and LHS. For the 10-th dimensional integral the best approach is produced by the optimized method OPT for $N = 3524578$ the

Table IX
ALGORITHM COMPARISON OF THE REs FOR THE 100-DIMENSIONAL INTEGRAL FOR DIFFERENT NUMBER OF POINTS.

# of points	OPT	t,s	FIBO	t,s	LHS	t,s	SOBOLS	t,s
2^{10}	5.18e-3	0.05	4.13e-1	0.06	5.18e-2	0.08	6.31e-2	18
2^{12}	3.18e-3	0.17	1.15e-1	0.18	3.22e-2	0.2	1.23e-2	34
2^{16}	1.44e-4	9.1	6.12e-2	9.2	8.32e-3	9.7	2.31e-3	170
2^{20}	6.38e-5	57.6	3.18e-2	58.7	4.51e-3	60	2.34e-4	861

Table X
ALGORITHM COMPARISON OF THE REs FOR THE 100-DIMENSIONAL INTEGRAL FOR A PRELIMINARY GIVEN TIME.

t, s	OPT	FIBO	LHS	SOBOLS
1	2.14e-3	7.18e-2	2.83e-2	9.31e-2
2	1.56e-3	6.02e-2	1.17e-2	8.66e-2
10	2.58e-4	4.12e-2	8.34e-3	6.94e-2
100	8.86e-6	1.13e-2	1.18e-3	3.88e-3

RE is $5.32e-8$ - see Table V and for 20s the best approach is again OPT - $9.13e-9$ in Table VI with two order better REs than FIBO and 3-4 order better REs than both SOBOLS and LHS. For the 30-th dimensional integral the best approach is produced by the optimized method OPT for $N = 1048576$ $8.81e-5$ - see Table VII and for 20s the best approach is again OPT - $2.33e-5$ in Table VIII with one order better REs than both SOBOLS and LHS and 3 order better REs than both FIBO, which shows that FIBO becomes inefficient for high dimensions. Finally, for the 100-th dimensional integral the best approach is produced by the optimized method OPT for $N = 2^{20}$ the RE is $6.38e-5$ - see Table IX and for 20s the best approach is OPT - $8.86e-6$ in Table X with 4 order better REs than FIBO and 3 orders better REs than SOBOLS and LHS. From all Tables we can conclude that the optimized lattice sequence OPT, used for the first time for the evaluation of this type of multidimensional integrals up to 100 dimensions, gives the best results compared to the other stochastic approaches with increasing the dimensionality of the multidimensional integral.

IV. CONCLUSION

In this paper an optimized lattice rule has been tested on multidimensional integrals reflated to neural networks up to 100 dimensions. A comprehensive experimental study of optimized lattice rule, Fibonacci lattice sets, Sobol scrambled sequence and Latin hypercube sampling has been done on some case test functions. Our approach is one of the best available algorithms for high dimensional integrals and the only possible methods, because the deterministic algorithms need an huge amount of time for the evaluation of the multidimensional integral, as it was discussed in this paper. At the same time the new method is suitable to deal with 100-dimensional problems for less than a minute on a laptop. It is an important element since this may be crucial in order to achieve a more reliable interpretation of the results in Bayesian statistics which is foundational in neural networks, artificial intelligence and machine learning.

REFERENCES

[1] N. Bahvalov (1959) On the Approximate Computation of Multiple Integrals, *Vestnik Moscow State University* **4**, 3–18.

[2] Centeno, V., Georgiev, I. R., Mihova, V., Pavlov, V. (2019, October). Price forecasting and risk portfolio optimization. In AIP Conference Proceedings (Vol. 2164, No. 1, p. 060006). AIP Publishing LLC.

[3] Dimov I., Monte Carlo Methods for Applied Scientists, New Jersey, London, Singapore, World Scientific, 2008, 291p.

[4] F.Y. Kuo and D. Nuyens (2016) Application of quasi-Monte Carlo methods to elliptic PDEs with random diffusion coefficients - a survey of analysis and implementation, *Foundations of Computational Mathematics* **16**(6), 1631–1696.

[5] Lin S., "Algebraic Methods for Evaluating Integrals in Bayesian Statistics," Ph.D. dissertation, UC Berkeley, May 2011.

[6] Lin, S., Sturmfels B., Xu Z.: Marginal Likelihood Integrals for Mixtures of Independence Models, Journal of Machine Learning Research, Vol. 10, pp. 1611-1631, 2009.

[7] Minasny B., McBratney B.: A conditioned Latin hypercube method for sampling in the presence of ancillary information Journal Computers and Geosciences archive, Volume 32 Issue 9, November, 2006, Pages 1378-1388.

[8] S.H. Paskov, *Computing high dimensional integrals with applications to finance*, Technical report CUCS-023-94, Columbia University (1994).

[9] Song, J., Zhao, S., Ermon, S., A-nice-mc: Adversarial training for mcmc. In Advances in Neural Information Processing Systems, pp. 5140-5150, 2017.

[10] Wang Y., Hickernell F., *An historical overview of lattice point sets*, 2002.

[11] Watanabe S., Algebraic analysis for nonidentifiable learning machines. NeuralComput.(13), pp. 899—933, April 2001.

[12] S. L. Zaharieva, I. Radoslavov Georgiev, A. N. Borodzhieva and V. Angelov Mutkov, "Classical Approach For Forecasting Temperature In Residential Premises Part 1," 2021 20th International Symposium Infoteh-Jahorina (infoteh), 2021, pp. 1-6.

An Optimized Stochastic Techniques related to Option Pricing

Venelin Todorov *†, Ivan Dimov†, Stefka Fidanova †, Stoyan Apostolov‡
*Institute of Mathematics and Informatics
Bulgarian Academy of Sciences
8 Acad. G. Bonchev Str., 1113 Sofia, Bulgaria
†Institute of Information and Communication Technologies
Bulgarian Academy of Sciences
25A Acad. G. Bonchev Str., 1113 Sofia, Bulgaria
‡Faculty of Mathematics and Informatics, Sofia University, Sofia 1126, Bulgaria
Email: vtodorov@math.bas.bg, venelin@parallel.bas.bg, ivdimov@bas.bg, stefka@parallel.bas.bg, stoyanrapostolov@gmail.com

***Abstract*—Recently stochastic methods have become very important tool for high performance computing of very high dimensional problems in computational finance. The advantages and disadvantages of the different highly efficient stochastic methods for multidimensional integrals related to evaluation of European style options will be analyzed. Multidimensional integrals up to 100 dimensions related to European options will be computed with highly efficient optimized lattice rules.**

I. Introduction

RECENTLY Monte Carlo (MC) and quasi-Monte Carlo (QMC) approaches have become a very attractive and necessary computational tools in finance [8]. The field of computational finance is becoming more complicated with increasing number of applications [2], [3]. The pricing of options is a very important in financial markets today and especially difficult when the dimension of the problem goes higher [2], [8], [9], [12]. MC and QMC methods are appropriate for solving multidimensional problems [4], since their computational complexity increases polynomially, but not exponentially with the dimensionality. MC methods are used not only for option pricing, but also in other problems in computational finance. The basic definitions and terminology used in the paper can be found in [1], [8].

The paper is organized as follows. The problem setting and motivation is presented in Section II. Some basic notations about the stochastic methods that we are going to use are presented in III. Numerical study and discussions are given in Section IV. The conclusions are given in Section V.

II. Problem settings and Motivation

Consider a European call option [8] whose payoff depends on $k > 1$ assets with prices $S_i, i = 1, ..., k.$. The payoff is the act or occasion of receiving money or material gain especially as compensation or as a bribe. Following [8] we assume that at expiry time T, and risk-free interest rate r, the payoff is given by $h(S'_1, \dots, S'_k)$, where S' denotes the value of the i-th asset at expiry. Then the value of the option satisfies:

$$V = e^{-r(T-t)}(2\pi(T-t))^{-k/2}(\det\Sigma)^{-1/2}(\sigma_1\dots\sigma_k)^{-1}$$

$$\int_0^\infty \cdots \int_0^\infty \frac{h(S'_1, \dots, S'_k)}{S'_1 \dots S'_k}$$

$$\exp\left(-0.5\alpha^\top\Sigma^{-1}\alpha\right) \mathrm{d}S'_1 \dots \mathrm{d}S^{'}_k,$$

$$\alpha_i = \left(\sigma_i(T-t)^{1/2}\right)^{-1}\left(\ln(S'_i/S_i) - (r - \sigma_i^2/2)(T-t)\right).$$

According to [8] the most important case in recent models is when the payoff function is the exponent function.

We will now give a brief explanation which demonstrates the strength of the MC and QMC approach. This is a case of practical high performance computations showing the high power and efficiency of the stochastic approach versus the deterministic one [4]. According to [4] we will choose 100 nodes on the each of the coordinate axes in the s-dimensional cube $G = E^s$ and we have to evaluate about 10^{100} values of the function $f(x)$. Assume a time of $10^{-7}s$ is necessary for calculating one value of the function [4]. So, a time of order 10^{93}s will be necessary for computation of the integral, and 1 year has 31536×10^3s.

Now MC approach [4] consists of generating N pseudo random values (points) (PRV) in G; in evaluating the values of $f(x)$ at these points; and averaging the computed values of the function. For each uniformly distributed random (UDR) point in G we have to generate 100 UDR numbers in $[0, 1]$. The probable error is estimated in [4]:

$$N \approx (\frac{0.6745||f||_{L_2}}{cM})^2 \times h^{-6}. \tag{1}$$

Venelin Todorov is supported by the National Scientific Program "Information and Communication Technologies for a Single Digital Market in Science, Education and Security (ICT in SES)", contract No DO1-205/23.11.2018, financed by the Ministry of Education and Science in Bulgaria and by the Bulgarian National Science Fund under Project DN 12/5-2017 "Efficient Stochastic Methods and Algorithms for Large-Scale Problems". Stoyan Apostolov is supported by the Bulgarian National Science Fund under Project KP-06-M32/2 - 17.12.2019 "Advanced Stochastic and Deterministic Approaches for Large-Scale Problems of Computational Mathematics". The work is also supported by the Project KP-06-Russia/17 "New Highly Efficient Stochastic Simulation Methods and Applications" funded by the National Science Fund - Bulgaria.

Assume that the expression in front of h^{-6} is of order 1 [4]. Here $h = 0.1$, and we have $N \approx 10^6$; so, it will be necessary to generate $100 \times 10^6 = 10 \times 10^7$ PRV. Usually, 2 operations are sufficient to generate a single PRV. According to [4] the time required to generate one PRV is the same as that for computation the value of $f(x)$. So, in order to solve the task with the same accuracy, a time of

$$10 \times 10^7 \times 2 \times 10^{-7} \approx 20s$$

will be necessary. We summarize that in the case of 100-dimensional integral it is 5×10^{91} times faster than the deterministic one. Also the stochastic approach is more accurate than the deterministic approach for higher dimensions. That motivates our study on the new highly efficient stochastic approaches for the problem under consideration.

III. Highly efficient stochastic approaches

We will make a brief description of the stochastic approaches that we are going to use in our survey. Such comparison, up to 100 dimensions, has been made for the first time for the problem under consideration.

A. Lattice rules

Fot the lattice point sets, please see cite[13], and more information can be found in the works of Sloan and Kachoyan [10], Sloan and Joe [11] and Hua and Wang [6].

We will use this rank-1 lattice sequence [13]:

$$\mathbf{x}_k = \left\{ \frac{k}{N}\mathbf{z} \right\}, \ k = 1, \dots, N, \tag{2}$$

where N is an integer, $N \geq 2$, $\mathbf{z} = (z_1, z_2, \dots z_s)$ is the generating vector and $\{z\}$ denotes the fractional part of z. For the definition of the $E_s^\alpha(c)$ and $P_\alpha(z, N)$ see [13].

The existence of lattice point sets with low discrepancy and low worst case error are closely connected [13].

While the theoretical result establish the existence of optimal generating vectors the main bottleneck lies in the creation of the optimal vectors, especially for very high dimensions [8].

The first generating vector in our study is the generalized Fibonacci numbers of the corresponding dimension:

$$\mathbf{z} = (1, F_n^{(s)}(2), \dots, F_n^{(s)}(s)). \tag{3}$$

where we use that

$$F_n^{(s)}(j) := F_{n+j-1}^{(s)} - \sum_{i=0}^{j-2} F_{n+i}^{(s)} \tag{4}$$

and $F_{n+l}^{(s)}$ ($l = 0, \dots, j-1, j$ is an integer, $2 \leq j \leq s$) is the term of the s-dimensional Fibonacci sequence [13].

Then each component of the generating vector $\mathbf{z}$ is defined by a sum of some terms of the generalized Fibonacci sequence with dimensionality s. For example:

$$F_n^{(s)}(2) = F_{n+1}^{(s)} - F_n^{(s)} = (F_n^{(s)} + F_{n-1}^{(s)} + \dots + F_{n-s+1}^{(s)}) - F_n^{(s)} = F_{n-1}^{(s)} + \dots + F_{n-s+1}^{(s)}.$$

Our generating vector (3) is transformed into [6], [13]:

$$\mathbf{z} = (1, F_{n-1}^{(s)} + F_{n-2}^{(s)} + \dots + F_{n-s+1}^{(s)}, \dots, F_{n-1}^{(s)} + F_{n-2}^{(s)}, F_{n-1}^{(s)}). \tag{5}$$

If we change the generating vector to be optimal in the way described in [7] we have improved the lattice sequence. This is a 200-dimensional base-2 generating vector of prime numbers for up to $2^{20} = 1048576$ points, constructed recently by Dirk Nuyens [7]. The special choice of this optimal generating vector is definitely more efficient than the Fibonacci generating vector, which is only optimal for the two dimensional case [13]. For this improved lattice rule, presented in the paper, is satisfied [7]:

$$D_N^* = \mathcal{O}\left(\frac{log^s N}{N}\right).$$

IV. Numerical examples and results

The numerical study includes high performance computing of the multidimensional integrals

$$I_s = \int\limits_{[0,1]^s} \exp\left(\prod_{i=1}^s x_i\right). \tag{6}$$

We will use the expansion of the exponential function in Taylor series and integrating $(x_1 \cdots x_s)^n$:

$$\int_{[0,1]^s} \exp\left(\prod_{i=1}^s x_i\right) =$$

$$= \sum_{n=0}^{\infty} \frac{1}{(n+1)^s n!} =_s F_s(1, \cdots, 1; 2, \cdots, 2; 1),$$

where ${}_pF_q(a_1, \cdots, a_p; b_1, \cdots, b_q; x)$ is the generalized hypergeometric function

$${}_pF_q(a_1, \cdots, a_p; b_1, \cdots, b_q; x) = \sum_{n=0}^{\infty} \frac{(a_1)_n \cdots (a_p)_n}{(b_1)_n \cdots (b_q)_n} \frac{x^n}{n!},$$

and $(c)_n = c(c+1) \cdots (c+n-1)$ is the Pochhammer symbol.

$$\int\limits_{[0,1]^3} exp(x_1 x_2 x_3) \approx 1.14649907. \tag{7}$$

$$\int\limits_{[0,1]^5} exp(\sum_{i=1}^5 0.5 a_i x_i^2 (2 + \sin \sum_{j=1, j\neq i}^5 x_j)) \approx 2.923651, \tag{8}$$

where $a_i = (1, 0.5, 0.2, 0.2, 0.2)$.

$$\int\limits_{[0,1]^8} exp(\sum_{i=1}^8 0.1 x_i) = 1.496805. \tag{9}$$

$$\int\limits_{[0,1]^{20}} exp(\prod_{i=1}^{20} x_i) \approx 1.00000949634. \tag{10}$$

We also have done high performance computing with our methods for the first time on a 100 dimensional integral:

$$I_{100} = \int_{[0,1]^{100}} \exp\left(\prod_{i=1}^{100} x_i\right). \qquad (11)$$

We calculate his reference value by using the exponential function in Taylor series and integrating $(x_1 \cdots x_{100})^n$ we receive

$$\int_{[0,1]^{100}} \exp\left(\prod_{i=1}^{100} x_i\right) =$$

$$= \sum_{n=0}^{\infty} \frac{1}{(n+1)^{100} n!} =_{100} F_{100}(1, \cdots, 1; 2, \cdots, 2; 1).$$

We also include in the experiments the 50-dimensional integral of the same kind:

$$I_{50} = \int_{[0,1]^{50}} \exp\left(\prod_{i=1}^{50} x_i\right). \qquad (12)$$

The results are given in the Tables including the relative error (RE) of the MC and QMC method that has been used, the CPU-time (T) in seconds and the number of realizations of the random variable (#). We will make a high performance computation, including the Optimized lattice rule (OP), the Fibonacci based rule (FI), the Adaptive approach (AD) and the Sobol quasi-random sequence (SO).

Table I
ALGORITHMIC COMPARISON OF RE FOR (7)

#	OP	T	AD	T	FI	T	SO	T
19513	1.93e-5	0.01	3.21e-4	2.21	4.69e-4	0.02	4.98e-5	0.56
35890	3.18e-6	0.04	6.55e-5	6.41	5.46e-6	0.06	1.56e-5	1.45
66012	2.65e-6	0.07	5.12e-5	9.86	5.34e-6	0.11	8.11e-6	2.31
121415	9.16e-7	0.12	5.11e-5	15.4	5.34e-6	0.12	3.08e-6	3.80
223317	8.01e-7	0.20	9.34e-5	24.2	1.73e-6	0.22	2.05e-6	6.13

Table II
ALGORITHMIC COMPARISON OF RE FOR THE (7)

T	OP	AD	FI	SO
0.1	9.16e-7	8.67e-4	1.32e-6	3.21e-4
1	6.37e-7	2.96e-5	3.22e-7	8.21e-5
2	4.22e-7	5.45e-4	2.06e-7	2.96e-5
5	1.84e-7	1.14e-4	1.47e-7	5.00e-6
10	6.09e-8	6.56e-5	3.89e-7	2.71e-6
20	1.57e-8	2.04e-5	1.53e-8	1.65e-6

Table III
ALGORITHMIC COMPARISON OF RE FOR THE (8)

#	OP	T	AD	T	FI	T	SO	T
13624	6.72e-5	0.02	1.89e-3	2.33	9.59e-4	0.03	1.76e-4	0.56
52656	1.53e-5	0.06	2.31e-3	6.18	6.96e-4	0.06	5.05e-5	1.45
103519	8.48e-6	0.09	2.01e-3	9.94	8.72e-5	0.13	2.70e-5	2.52
203513	6.25e-6	0.15	3.42e-4	16.2	8.04e-5	0.25	7.57e-6	6.07
400096	8.16e-7	0.40	9.12e-4	45.6	7.26e-5	0.50	2.52e-6	10.63

Table IV
ALGORITHMIC COMPARISON OF RE FOR THE (8)

T	OP	AD	FI	SO
0.1	3.07e-6	1.34e-2	7.26e-5	8.22e-4
1	1.32e-6	2.44e-3	2.28e-5	2.91e-4
5	1.13e-6	4.93e-4	5.94e-6	1.71e-5
10	5.47e-7	1.88e-3	3.85e-7	1.79e-5
20	3.52e-7	2.71e-4	7.49e-7	4.71e-6

Table V
ALGORITHMIC COMPARISON OF RE FOR THE (9)

#	OP	T	AD	T	FI	T	SO	T
16128	1.79e-6	0.04	1.10e-5	12.6	8.08e-4	0.03	8.87e-5	0.13
32192	1.56e-6	0.05	3.32e-5	33.3	1.03e-4	0.07	5.42e-5	0.58
64256	8.01e-7	0.08	4.65e-5	54.2	5.03e-5	0.11	2.34e-5	2.49
128257	6.22e-7	0.13	8.25e-6	88.3	8.13e-6	0.14	4.45e-6	6.36
510994	3.21e-7	0.34	7.07e-6	233.6	5.95e-6	0.57	3.32e-6	19.45

Table VI
ALGORITHMIC COMPARISON OF RE FOR THE (9)

T	OP	AD	FI	SO
1	2.18e-7	6.34e-4	5.34e-6	2.02e-5
2	1.32e-7	1.58e-4	2.57e-6	2.73e-5
5	9.03e-8	1.44e-4	1.52e-7	8.88e-6
10	5.00e-8	6.61e-5	3.45e-6	5.23e-6
20	2.55e-8	2.77e-5	1.82e-7	2.11e-6

Table VII
ALGORITHMIC COMPARISON OF RE FOR THE (10)

#	OP	T	AD	T	FI	T	SO	T
2048	2.84e-6	0.02	1.14e-2	8.6	8.22e-5	0.03	8.44e-4	0.13
16384	1.04e-6	0.12	4.96e-4	60.3	3.12e-5	0.13	6.82e-5	1.68
65536	9.21e-7	0.91	9.75e-4	474.2	1.36e-5	1.17	8.34e-6	8.69
131072	6.15e-7	2.13	1.25e-5	888.3	8.85e-6	2.34	3.77e-6	14.36
524288	5.33e-8	8.13	1.96e-6	2356	2.15e-6	8.34	1.91e-7	57

Table VIII
ALGORITHMIC COMPARISON OF RE FOR THE (10)

T	OP	AD	FI	SO
1	9.14e-7	1.58e-3	1.48e-5	3.25e-5
2	1.08e-7	1.028e-3	9.17e-6	3.97e-5
5	5.87e-8	8.58e-4	5.19e-6	1.45e-5
10	3.56e-8	4.31e-4	1.73e-6	2.71e-6
20	1.23e-8	1.27e-4	1.38e-7	1.76e-6

Table IX
ALGORITHMIC COMPARISON OF RE FOR THE (12)

#	OP	T	FI	T	SO	T
2^{10}	7.88e-6	0.05	6.23e-4	0.08	8.88e-5	3.5
2^{12}	1.88e-6	0.17	1.55e-4	0.35	5.21e-5	16
2^{16}	8.44e-8	2.14	9.72e-5	5.21	9.11e-4	73
2^{20}	4.28e-8	17.65	6.08e-5	32.76	4.88e-6	276

For the 3-dimensional integral, for the number of samples

Table X
ALGORITHMIC COMPARISON OF RE FOR THE (12)

T	OP	FI	SO
1	9.14e-7	1.58e-3	1.48e-4
2	7.51e-7	1.028e-3	9.17e-5
10	9.34e-8	3.01e-4	8.73e-5
100	1.34e-9	5.23e-5	1.03e-5

Table XI
ALGORITHM COMPARISON OF THE RE FOR THE (11)

#	OP	T	FI	T	SO	T
2^{10}	6.83e-3	0.05	4.13e-1	0.06	6.31e-2	18
2^{12}	3.77e-4	0.17	1.15e-1	0.18	1.23e-2	34
2^{16}	3.36e-5	9.1	6.12e-2	9.2	2.31e-3	170
2^{20}	4.78e-6	57.6	3.18e-2	58.7	2.34e-4	861

Table XII
ALGORITHM COMPARISON OF THE RE FOR THE 100-DIMENSIONAL INTEGRAL (11)

T	OP	FI	SO
1	2.67e-3	7.18e-2	9.31e-2
2	1.89e-4	6.02e-2	8.66e-2
10	3.22e-5	4.12e-2	6.94e-2
100	8.16e-7	1.13e-2	3.88e-3

Generalized Fibonacci numbers of the correspond-ing dimensionality, the best relative error is produced by the optimized lattice algorithm OPT - see Table I, but for a preliminary given time in seconds the optimized method OPT and the Fibonacci latice rule FIBO gives results of the same order - see Table II. For the 5-dimesnional integral again the best approach is OPT method, for $N = 440096$ it gives relative error of $8.16e-7$ - see Table III, while for $20s$ again FIBO method gives results of the same order as the optimized method - see Table IV. For the 8-dimensional integral the Adaptive approach, the Sobol QMC algorithm and the Fibonacci approach produce relative error of the same order - see Table V, but for a preliminary given time in seconds, Fibonacci approach is better than both Sobol QMC and Adaptive approach - see Table VI. For the 20-dimensional integral Sobol QMC approach is better than both Fibonacci and Adaptive approach - see Table VII and Adaptive approach requires very huge amount of time - near one hour for number of samples $N = 524888$ due to the division of the subareas in the description of the algorithm. Thats why we omit this algorithm for the 50 and 100-dimensional integrals. For $20s$ for 20-dimensional integral the best result is produced again by the optimized lattice rule - $1.23e-8$ in Table VIII. For the 50-dimensional integral Fibonacci approach is worse than Sobol approach by at least 1 order - see Table IX, but for a preliminary given time in seconds Sobol QMC and Fibonacci approach give relative errors of the same order - see Table X. It is worth mentioning that the SOBOL approach requires more amount of time due to generation of the sequence, while Fibonacci lattice rules and Optimized approach are more faster and computationally efficient algorithms. For the 100-dimensional integral the best result is produced by the optimized lattice approach - it gives $4.78e-6$ for number of samples $N = 2^{20}$ - see Table XI and for $100s$ it produces a relative error of $8.16e-7$ which is very high accuracy and with 3 to 4 orders better than the other stochastic approaches. So we demonstrate here the advantages of the new lattice method and its capability to achieve very high accuracy for less than a minute on a laptop with a quad core CPU.

V. CONCLUSION

A comprehensive experimental study of optimized lattice rule, Fibonacci lattice sets, Sobol sequence and Adaptive approach has been done for the first time on some case test functions related to option pricing. Optimized lattice rule described here is not only one of the best available algorithms for high dimensional integrals but also one of the few possible methods, because in this work we show that the deterministic algorithms need a huge amount of time for the evaluation of the multidimensional integral, as it was discussed in this paper. The numerical tests show that the improved lattice rule is efficient for multidimensional integration and especially for computing multidimensional integrals of a very high dimensions up to 100. The novelty is that the new proposed optimized method gives very high accuracy for less than a minute on laptop even for 100-dimensional integral. It is an important element since this may be crucial in order to achieve a more reliable interpretation of the results in European style options which is foundational in computational finance.

REFERENCES

[1] N. Bakhvalov (2015) On the approximate calculation of multiple integrals, *Journal of Complexity* **31**(4), 502–516.

[2] P.P. Boyle, Y. Lai and K. Tan, *Using lattice rules to value low-dimensional derivative contracts* (2001).

[3] Centeno, V., Georgiev, I. R., Mihova, V., & Pavlov, V. (2019, October). Price forecasting and risk portfolio optimization. In AIP Conference Proceedings (Vol. 2164, No. 1, p. 060006). AIP Publishing LLC.

[4] Dimov I., Monte Carlo Methods for Applied Scientists, New Jersey, London, Singapore, World Scientific, 2008, 291p.

[5] Hua, L.K. and Wang, Y., *Applications of Number Theory to Numerical analysis*, 1981.

[6] L. K. Hua and Y. Wang, *Applications of number theory to numerical analysis*, (Springer 1981).

[7] F.Y. Kuo and D. Nuyens (2016) Application of quasi-Monte Carlo methods to elliptic PDEs with random diffusion coefficients - a survey of analysis and implementation, *Foundations of Computational Mathematics* **16**(6), 1631–1696.

[8] Y. Lai and J. Spanier, *Applications of Monte Carlo/Quasi-Monte Carlo methods in finance: option pricing*, Proceedings of the Claremont Graduate University conference (1998).

[9] S.H. Paskov, *Computing high dimensional integrals with applications to finance*, Technical report CUCS-023-94, Columbia University (1994).

[10] I.H. Sloan and P.J. Kachoyan (1987) Lattice methods for multiple integration: Theory, error analysis and examples, *SIAM J. Numer. Anal.* **24**, 116–128.

[11] I.H. Sloan and S. Joe, Lattice Methods for Multiple Integration, *Lattice methods for multiple Integration*, (Oxford University Press 1994).

[12] S. L. Zaharieva, I. Radoslavov Georgiev, V. A. Mutkov and Y. Branimirov Neikov, ”Arima Approach For Forecasting Temperature In A Residential Premises Part 2,” 2021 20th International Symposium infoteh-jahorina (Infoteh), 2021, pp. 1-5.

[13] Y. Wang and F. J. Hickernell (2000) *An historical overview of lattice point sets*, in MC and QMC Methods 2000, Proceedings of a Conference held at Hong Kong Baptist University, China.

DOI: 10.15439/2021F77
ISSN 2300-5963 ACSIS, Vol. 25

Interval-Valued Intuitionistic Fuzzy Decision-Making Method using Index Matrices and Application in Outsourcing

Velichka Traneva
Prof. Asen Zlatarov University
1 Prof. Yakimov Blvd, Burgas 8000, Bulgaria
Email: veleka13@gmail.com

Stoyan Tranev
Prof. Asen Zlatarov University
1 Prof. Yakimov Blvd, Burgas 8000, Bulgaria
Email: tranev@abv.bg

Deyan Mavrov
Prof. Asen Zlatarov University
1 Prof. Yakimov Blvd, Burgas 8000, Bulgaria
Email: dg@mavrov.eu

***Abstract*—Selecting a suitable outsourcing service provider is a challenging problem that requires discussion among a group of experts. The problems of this type belongs to the area of multicriteria decision-making. Interval-valued intuitionistic fuzzy sets, which are an extension of intuitionistic fuzzy sets, are a capable tool in modeling uncertain problems. In this paper we will formulate an optimal interval-valued intuitionistic fuzzy multicriteria decision-making problem in outsourcing and propose a new approach for the selection of the most appropriate candidates; as well as a software program for its automated solution, based on our previous libraries. As an example of a case study, an application of the algorithm on real data from a refinery is demonstrated.**

I. Introduction

The AIM of multi-criteria decision-making (MCDM) is to determine an optimal alternative having the highest degree of desirability with respect to all relevant goals [3]. Most decisions are not made on the basis of exact data. Zadeh's Fuzzy Logic [15] has emerged to help model this vague environment. The uncertainty in the MCDM-problem may be caused by unavailable or indeterminate characteristics of the alternative options or from the inability of the experts to formulate a precise evaluation [16]. Atanassov, in 1983, introduced the notion of an intuitionistic fuzzy set (IFS, [5]) as a generalization of fuzzy sets, which adds a degree of hesitance. Later, Atanassov and Gargov proposed in 1989 the concept of interval-valued intuitionistic fuzzy sets (IVIFSs, [7]). There are many papers for application of IVIF theory in MCDM-problems.

In this study, an optimal generalized MCDM-approach (IVIFIMOA) for selecting the most appropriate outsourcing providers will be formulated over IVIF data. In [19], [20] we have proposed an IF algorithm for the selection of outsourcing service providers using the concepts of IMs [4] and IF logic [5]. Here we will extend this approach to interval-valued IF (IVIF) logic [7].

The rest of the paper contains the following sections: Section 2 presents IVIFSs and IMs. Section 3 formulates an optimal IVIF problem for the selection of outsourcing providers, gives an algorithm for its solution and describes the software implementation. A real life case study is described. Section 4 concludes the work and gives future suggestions.

II. Basic concepts of IMs and IVIF logic

This section recalls some basic concepts on interval-valued intuitionistic fuzzy pairs (IVIFPs) from [6], [12] and on the index matrix apparatus from [4], [8], [23].

A. Interval-Valued Intuitionistic Fuzzy Logic

The concept of IVIFPs was introduced in [12]. The **IVIFP** is an object of the form $\langle M,N\rangle$, where $M,N \subseteq [0,1]$ are closed sets, $M=[\inf M,\sup M]\,, N=[\inf N,\sup N]$ and

$$\sup M+\sup N \leq 1,$$

that is used as an evaluation of some object or process and whose components (M and N) are interpreted as intervals of degrees of membership and non-membership, or intervals of degrees of validity and non-validity, etc.

Let us have two IVIFPs $x=\langle M,N\rangle$ and $y=\langle P,Q\rangle$. In [6], [12] are defined the operations classical negation, conjunction, disjunction, multiplication with constant, and difference.

The forms of the relations with IVIFPs are given in [6].

B. Interval-Valued Intuitionistic Fuzzy Index Matrices

Let $\mathscr{I}$ be a fixed set. Three-dimensional interval-valued intuitionistic fuzzy index matrix (3-D IVIFIM) with index sets K,L and H $(K,L,H \subset \mathscr{I})$, we denote the object [4], [8]:

$$[K,L,H,\{\langle M_{k_i,l_j,h_g},N_{k_i,l_j,h_g}\rangle\}]$$

Work is supported by the Asen Zlatarov University through project Ref. No. NIX-440/2020 "Index matrices as a tool for knowledge extraction".

$$\equiv \begin{array}{c|ccc} h_g \in H & l_1 & \dots & l_n \\ \hline k_1 & \langle M_{k_1,l_1,h_g}, M_{k_1,l_1,h_g}\rangle & \dots & \langle M_{k_1,l_n,h_g}, N_{k_1,l_n.h_g}\rangle \\ \vdots & \vdots & \ddots & \vdots \\ k_m & \langle M_{k_m,l_1,h_g}, N_{k_m,l_1,h_g}\rangle & \dots & \langle M_{k_m,l_n,h_g}, N_{k_m,l_n,h_g}\rangle \end{array}, \quad (1)$$

where for every $1 \le i \le m, 1 \le j \le n, 1 \le g \le f$:

$$M_{k_i,l_j,h_g} \subseteq [0,1], N_{k_i,l_j,h_g} \subseteq [0,1], \sup M_{k_i,l_j,h_g} + \sup N_{k_i,l_j,h_g} \le 1.$$

Over every two 3-D IVIFIMs A and B we can apply the operations addition, transposition, multiplication, projection and substitution, as defined in [4], [8], [21]. [22] defines the operation "aggregation by one dimension".

A Level operator for decreasing the number of elements of IVIFM: Let $\langle \alpha, \beta \rangle$ is an IVIFP, then according to [14] $N^{>}_{\alpha,\beta}(A) = [K, L, H, \{\langle R_{k_i,l_j,h_g}, P_{k_i,l_j,h_g}\rangle\}]$, where

$$\langle R_{k_i,l_j,h_g}, P_{k_i,l_j,h_g}\rangle$$

$$= \begin{cases} \langle M_{k_i,l_j,h_g}, N_{k_i,l_j,h_g}\rangle & \text{if } \langle R_{k_i,l_j,h_g}, P_{k_i,l_j,h_g}\rangle > \langle \alpha, \beta\rangle \\ \langle [0,0],[1,1]\rangle & \text{otherwise} \end{cases} \quad (2)$$

III. Optimal Interval-Valued Intuitionistic Fuzzy Selection for the Outsourcing Service Providers

Here, we will formulate an optimal IVIF outsourcing problem.

The management team of a company has selected the following activities $v_e (1 \le e \le u)$ to be offered for outsourcing in order to increase the profitability of the enterprise. An expert team, consisting of experts $\{r_1, \dots, r_s, \dots, r_D\}$ has proposed an evaluation system, giving each candidate $\{k_1, \dots, k_i, \dots, k_m\}$ (for $i = 1,...,m$) for the respective outsourced service $v_e (1 \le e \le u)$, an evaluation by each criterion $\{c_1, \dots, c_j, \dots, c_n\}$ (for $j = 1,...,n$). The weight coefficients of each assessment criteria c_j (for $j = 1,...,n$) according to their priority for the service v_e are given in the form of IVIFPs - pk_{c_j,v_e} (for $j = 1,...,n$). Each expert has an IVIFP rating $r_s = \langle \varDelta_s, \varepsilon_s \rangle$ $(1 \le s \le D)$. Let the number of his/her own participations in previous outsourcing procedures be equal to $\varGamma_s (s = 1,...,D)$, respectively. All applicants need to be evaluated by the team of experts according to the established criteria in the company at the current time point h_f for their application for each outsourced service $v_e (1 \le e \le u)$, and their evaluations ev_{k_i,c_j,d_s} (for $1 \le i \le m, 1 \le j \le n, 1 \le s \le D$) are IVIFPs. Now we need to find the optimal assignment of candidates.

A. Optimal IVIF Selection of the Providers

To solve this problem, we propose a new approach - IVIFIMOA, described with mathematical notation and pseudocode:

Step 1. This step creates an expert 3-D evaluation IM EV. It is possible for the experts to include assessments for the same candidates from a previous evaluation IM at time points $h_1, ..., h_g, ..., h_{f-1}$. The team of experts needs to evaluate the candidates for the services according to the approved criteria in the company at the current time moment h_f. The experts are uncertain about their evaluations due to changes in some uncontrollable factors. The evaluations are IVIFPs.

It is possible that some of the experts' assessments are incorrect from an IVIF point of view. In [7], different ways for altering incorrect experts' estimations are discussed. Let us propose that, the estimations of the $D_s (1 \le s \le D)$ expert are correct and described by the IVIFIM $EV_s = [K, C, H, \{ev_{k_i,c_j,d_s,h_g}\}]$ as follows:

$$\begin{array}{c|ccc} h_g \in H & c_1 & \dots & c_n \\ \hline k_1 & \langle M_{k_1,c_1,d_s,h_g}; N_{k_1,c_1,d_s,h_g}\rangle & \dots & \langle M_{k_1,c_n,d_s,h_g}; N_{k_1,c_n,d_s,h_g}\rangle \\ \vdots & \vdots & \ddots & \vdots \\ k_m & \langle M_{k_m,c_1,d_s,h_g}; N_{k_m,c_1,d_s,h_g}\rangle & \dots & \langle M_{k_m,c_n,d_s,h_g}; N_{k_m,c_n,d_s,h_g}\rangle \end{array}, \quad (3)$$

where $K = \{k_1, k_2, \dots, k_m\}$, $C = \{c_1, c_2, \dots, c_n\}, H = \{h_1, h_2, \dots, h_f\}$ and IVIFP $\{ev_{k_i,c_j,d_s,h_g}\}$ is the estimate of the d_s-th expert for the k_i-th candidate by the c_j-th criterion at a moment h_g.

Let us apply the α_H-th aggregation operation $\alpha_{EV_s,\#_q}$ to find the evaluation of the d_s-th expert $(s = 1,...,D)$, where $1 \le q \le 3$. We get the 3-D IVIFIM $EV[K, C, E, \{ev_{k_i,c_j,d_s}\}]$ with the evaluations of all experts for all candidates:

$$EV = \alpha_{EV_1,\#_q}(H, d_1) \oplus_{(max,min)} \oplus_{(max,min)} \cdots$$

$$\cdots \oplus_{(max,min)} \alpha_{EV_D,\#_q}(H, d_D) \quad (4)$$

Go to *Step 2.*

Step 2. Let the score (rating) r_s of the d_s-th expert $(d_s \in E)$ be specified by an IVIFP $\langle \delta_s, \varepsilon_s \rangle$. δ_s and ε_s are interpreted respectively as his degree of competence and of incompetence. Then we create $EV^*[K, C, E, \{ev^*{}_{k_i,c_j,d_s}\}]$:

$$EV^* = r_1 pr_{K,C,d_1} EV \dots \oplus_{(max,min)} r_D pr_{K,C,d_D} EV; \quad (5)$$

$EV := EV^*(ev_{k_i,l_j,d_s} = ev^*_{k_i,l_j,d_s}, \ \forall k_i \in K, \forall l_j \in L, \forall d_s \in E)$.

Then α_E-th aggregation operation is applied to find the aggregated assessment $R = \alpha_{E,\#_q}(EV, h_f)$ $(1 \le q \le 3)$ of the k_i-th candidate against the c_j-th criterion at the moment $h_f \notin E$. If q is 2 or 3, then the evaluation of the candidates is more optimistic as outsorcing service provider. Go to *Step 3.*

Step 3. Let us define the 3-D IFIM $PK[C, V, h_f, \{pk_{c_j,v_e,h_f}\}]$ of the weight coefficients of the assessment criterion according to its priority to the outsourcing service v_e $(1 \le e \le u)$, where $C = \{c_1, \dots, c_n\}$, $V = \{v_1, \dots, v_u\}$ and all elements pk_{c_j,v_e,h_f} are IVIFPs. The transposed IM of R is founded under the form $R^T[K, C, h_f]$ and is calculated 3-D IVIFIM

$$B[K, V, h_f, \{b_{k_i,v_e,h_f}\}] := R^T \odot_{(\circ,*)} PK, \quad (6)$$

which contains the cumulative estimates of the k_i-th candidate (for $1 \le i \le m$) for the v_e-th outsourcing service. If a candidate $k_i (1 \le i \le m)$ does not wish to participate in the competition to provide an outsourcing service v_e, then the element b_{k_i,v_e,h_f} is equal to $\langle [0,0],[1,1]\rangle$. Go to *Step 4.*

Step 4. The aggregation operation $\alpha_{K,\#_q}(B, k_0)$ is applied by the dimension K to find the most suitable candidate for the outsourcing service v_e, where $k_0 \notin K, 1 \le q \le 3$.

If the company requires a different candidate for each service, then it is necessary to apply the IVIF Hungarian algorithm [18] to the data contained in the IVIFIM B and then the optimal allocation of the candidates will be found. It is possible to reduce the candidates with an overall score lower than the IVIFP $\langle \alpha, \beta \rangle$ applying the level-operator (2) to IVIFIM B before the algorithm is implemented. Go to *Step 5.*

Step 5. At this step of the algorithm, we need to determine whether there are correlations between some of the evaluation criteria [11]. The procedure of IVIF-form of ICrA (IVIFICrA), based on the intercriteria analysisis [10] is discussed in [13].

Let IVIFP $\langle \alpha, \beta \rangle$ be given. The criteria C_k and C_l are in: strong (α, β)-positive consonance, if $\inf M_{C_k,C_l} > \alpha$ and $\sup N_{C_k,C_l} < \beta$; weak (α, β)-positive consonance, if $\sup M_{C_k,C_l} > \alpha$ and $\inf N_{C_k,C_l} < \beta$; strong (α, β)-negative consonance, if $\sup M_{C_k,C_l} < \alpha$ and $\inf N_{C_k,C_l} > \beta$; weak (α, β)-negative consonance, if $\inf M_{C_k,C_l} < \alpha$ and $\sup N_{C_k,C_l} > \beta$; (α, β)-dissonance, otherwise.

After application of the IVIFICrA over IFIM R we determine which criteria are in consonance. Then, we can evaluate their complexity and more expensive or slower criteria can be removed from the evaluation system. If $O = \{O_1, ..., O_V\}$ are the criteria that can be omitted, then we can reduce R by IM-operation $R* = R_{(O,\perp)}$. Go to *Step 6.*

Step 6. The last step determinates the new rating coefficients of the experts. Let the expert d_s $(s = 1, ..., D)$ participate in Γ_s procedures, on the basis of which his score $r_s = \langle \Delta_s, \varepsilon_s \rangle$ is determined, then after his participation in $(\Gamma_s + 1)$-th procedure his score will be determined by [5]:

$$\langle \Delta_s^{'}, \varepsilon_s^{'} \rangle = \begin{cases} \langle [\frac{\inf \Delta.\Gamma+1}{\Gamma+1}, \frac{\sup \Delta.\Gamma+1}{\Gamma+1}], [\frac{\inf \varepsilon.\Gamma}{\Gamma+1}, \frac{\sup \varepsilon.\Gamma}{\Gamma+1}] \rangle, \\ \text{(if the expert's estimation is correct)} \\ \langle [\frac{\inf \Delta.\Gamma}{\Gamma+1}, \frac{\sup \Delta.\Gamma}{\Gamma+1}], [\frac{\inf \varepsilon.\Gamma}{\Gamma+1}, \frac{\sup \varepsilon.\Gamma}{\Gamma+1}] \rangle, \\ \text{(if the expert had not given any estimation)} \\ \langle [\frac{\inf \Delta.\Gamma}{\Gamma+1}, \frac{\sup \Delta.\Gamma}{\Gamma+1}], [\frac{\inf \varepsilon.\Gamma+1}{\Gamma+1}, \frac{\sup \varepsilon.\Gamma+1}{\Gamma+1}] \rangle, \\ \text{(if the expert's estimation is incorrect)} \end{cases} \quad (7)$$

The complexity of the algorithm whithout step 5 is $O(Dmn)$ (the complexity of the ICrA in the step 5 is $O(m^2n^2)$ [17]).

In order to apply IVIFIMOA algorithm on real data more easily, we are currently developing a command line utility. It is written in C++ and uses an IM template class (*IndexMatrix⟨T⟩*), which implements the basic IM operations [2]. Any type used with the IM class must provide methods for performing operations on the current object and between two objects, so that they can be substituted in the already prepared IM operation methods. As part of a previous work on intuitionistic fuzzy ANOVA [24], we delevoped a class representing IFPs. Using the work done previously, for this project we are developing a class for IVIF pairs.

B. Real life case study

In this section, the proposed IVIFIMOA approach is applied to a real case study in an oil refinery [1] with the help of the "IVIFIMOA" software utility. The studied refinery adopts the outsourcing model. After the restructuring, the following activities remain outside the company, and will be offered for outsourcing: v_1 - trade and distribution of high quality fuels, polymers and petrochemicals; v_2 - engineering activity, specialized in consulting, preparation of technical and economic opinions, detailed projects with author's supervision; v_3 - transport service for public transport of goods and passengers, as well as services with construction machinery; v_4 - aviation fuel distributor. For this purpose, the refinery invites a team of the experts d_1, d_2 and d_3 to evaluate the candidates k_i (for $1 \le i \le 4$) for the outsourced refinery services. The real evaluation system of outsourcing providers selection is determined on the basis of 5 criteria as follows: C_1 - compliance of the outsourcing service provider with its corporate culture; C_2 - understanding of the outsourcing service by the provider; C_3 - necessary resources of the outsourcing provider for the implementation of the service; C_4 - price of the provided service; C_5 - opportunity for strategic development of the outsourcing service together with the outsourcing-assignor.

The weight coefficients for the service v_e - pk_{c_j,v_e} for the criteria c_j (for $j = 1, ..., 5$) according to their priority for the service v_e $(e = 1, 2, 3, 4)$ and the ratings of the experts $\{r_1, r_2, r_3\}$ be given under the form of IVIFPs. The aim of the problem is to optimally select the outsourcing providers.

An optimal solution of the problem:

Step 1. A 3-D expert evaluation IVIFIM $EV[K, C, E, \{es_{k_i,c_j,d_s}\}]$ is created and IVIFP $\{ev_{k_i,c_j,d_s}\}$ (for $1 \le i \le 4, 1 \le j \le 5, 1 \le s \le 3$) is the estimate of the d_s-th expert for the k_i-th candidate by the c_j-th criterion

$$\left\{ \begin{array}{c|cc} d_1 & c_1 & c_2 \\ \hline k_1 & \langle [0.4, 0.5], [0.3; 0.4] \rangle & \langle [0.7, 0.8], [0.1, 0.2] \rangle \\ k_2 & \langle [0.7, 0.8], [0.1, 0.2] \rangle & \langle [0.5, 0.6], [0.1, 0.2] \rangle \\ k_3 & \langle [0.5, 0.6], [0.2, 0.3] \rangle & \langle [0.6, 0.7], [0.1, 0.2] \rangle \\ k_4 & \langle [0.6, 0.7], [0.0, 0.1] \rangle & \langle [0.7, 0.8], [0.1, 0.2] \rangle \end{array} \right.$$

$$\begin{array}{ccc} c3 & c_4 & c_5 \\ \hline \langle [0.3, 0.4], [0.2, 0.3] \rangle & \langle [0.5, 0.6], [0.1, 0.2] \rangle & \langle [0.6, 0.7], [0.0, 0.1] \rangle \\ \langle [0.5, 0.6], [0.2, 0.3] \rangle & \langle [0.3, 0.4], [0.4, 0.5] \rangle & \langle [0.7, 0.8], [0.0, 0.1] \rangle \\ \langle [0.6, 0.7], [0.2, 0.3] \rangle & \langle [0.2, 0.3], [0.4, 0.5] \rangle & \langle [0.6, 0.7], [0.1, 0.2] \rangle \\ \langle [0.8, 0.9], [0.0, 0.1] \rangle & \langle [0.4, 0.5], [0.3, 0.4] \rangle & \langle [0.5, 0.6][0.2, 0.3] \rangle \end{array},$$

$$\begin{array}{c|cc} d_2 & c_1 & c_2 \\ \hline k_1 & \langle [0.3, 0.4], [0.2, 0.3] \rangle & \langle [0.9, 1], [0.1, 0.0] \rangle \\ k_2 & \langle [0.6, 0.7], [0.1, 0.2] \rangle & \langle [0.4, 0.5], [0.2, 0.3] \rangle \\ k_3 & \langle [0.4, 0.5], [0.3, 0.4] \rangle & \langle [0.7, 0.8], [0.0, 0.1] \rangle \\ k_4 & \langle [0.5, 0.6], [0.0, 0.1] \rangle & \langle [0.6, 0.7], [0.1, 0.2] \rangle \end{array}$$

$$\begin{array}{ccc} c3 & c_4 & c_5 \\ \hline \langle [0.4, 0.5], [0.3, 0.4] \rangle & \langle [0.4, 0.5], [0.0, 0.1] \rangle & \langle [0.5, 0.6], [0.1, 0.2] \rangle \\ \langle [0.4, 0.5], [0.0, 0.1] \rangle & \langle [0.4, 0.5], [0.3, 0.4] \rangle & \langle [0.6, 0.7], [0.0, 0.1] \rangle \\ \langle [0.5, 0.6], [0.2, 0.3] \rangle & \langle [0.2, 0.3], [0.5, 0.6] \rangle & \langle [0.7, 0.8], [0.1, 0.2] \rangle \\ \langle [0.7, 0.8], [0.0, 0.1] \rangle & \langle [0.5, 0.6], [0.1, 0.2] \rangle & \langle [0.3, 0.4], [0.2, 0.3] \rangle \end{array},$$

$$\begin{array}{c|cc} d_3 & c_1 & c_2 \\ \hline k_1 & \langle [0.5, 0.6], [0.3, 0.4] \rangle & \langle [0.6, 0.7], [0.1, 0.2] \rangle \\ k_2 & \langle [0.7, 0.8], [0.0, 0.1] \rangle & \langle [0.6, 0.7], [0.1, 0.2] \rangle \\ k_3 & \langle [0.6, 0.7], [0.1, 0.2] \rangle & \langle [0.7, 0.8], [0.1, 0.2] \rangle \\ k_4 & \langle [0.8, 0.9], [0.0, 0.1] \rangle & \langle [0.4, 0.5], [0.3, 0.4] \rangle \end{array}$$

$$\left. \begin{array}{ccc} c_3 & c_4 & c_5 \\ \hline \langle [0.2, 0.3], [0.3, 0.4] \rangle & \langle [0.2, 0.3], [0.5, 0.6] \rangle & \langle [0.6, 0.7], [0.1, 0.2] \rangle \\ \langle [0.8, 0.7], [0.0, 0.1] \rangle & \langle [0.1, 0.2], [0.4, 0.5] \rangle & \langle [0.4, 0.5], [0.0, 0.1] \rangle \\ \langle [0.7, 0.8], [0.0, 0.1] \rangle & \langle [0.2, 0.3], [0.3, 0.4] \rangle & \langle [0.7, 0.8], [0.1, 0.2] \rangle \\ \langle [0.6, 0.7], [0.0, 0.1] \rangle & \langle [0.6, 0.7], [0.1, 0.2] \rangle & \langle [0.4, 0.5], [0.2, 0.3] \rangle \end{array} \right\}$$

The experts' evaluations are transformed from positive integers to IVIF data using the method described in [7].

Step 2. Let the experts have the following rating coefficients respectively: $\{r_1, r_2, r_3\} = \{\langle [0.7, 0.8], [0.0, 0.1] \rangle,$ $\langle [0.6, 0.7], [0.0, 0.1] \rangle, \langle [0.8, 0.9], [0.0, 0.1] \rangle\}$.

We create $EV^*[K, C, E, \{ev^*\}] = r_1 pr_{K,C,d_1} EV \oplus_{(max,min)} r_2 pr_{K,C,d_2} EV \oplus_{(max,min)} r_3 pr_{K,C,d_3} EV$. Then, $EV := EV^*$.

Let us apply the optimistic aggregation operation $\alpha_{E,(\max,\min)}(EV, h_f) = R[K, h_f, C]$ to find the aggregated value of the k_i-th candidate against the c_j-th criterion in a current time-moment $h_f \notin D$.

Step 3. The 3-D IFIM $PK[C, V, h_f, \{pk_{c_j,v_e,h_f}\}]$ of the weight coefficients of the assessment criterion according to its priority to the service $v_e(e=1,2,3,4)$ has the following form:

$$\begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_4 \\ \hline c_1 & \langle[0.8,0.9],[0.0,0.1]\rangle & \langle[0.7,0.8],[0.1,0.2]\rangle & \langle[0.5,0.6],[0.1,0.2]\rangle & \langle[0.6,0.7],[0.1,0.2]\rangle \\ c_2 & \langle[0.7,0.8],[0.0,0.1]\rangle & \langle[0.5,0.6],[0.1,0.2]\rangle & \langle[0.6,0.7],[0.0,0.2]\rangle & \langle[0.7,0.8],[0.1,0.2]\rangle \\ c_3 & \langle[0.5,0.6],[0.1,0.2]\rangle & \langle[0.8,0.9],[0.0,0.1]\rangle & \langle[0.4,0.5],[0.2,0.3]\rangle & \langle[0.6,0.7],[0.1,0.2]\rangle \\ c_4 & \langle[0.8,0.9],[0.0,0.1]\rangle & \langle[0.8,0.9],[0.0,0.1]\rangle & \langle[0.7,0.8],[0.0,0.1]\rangle & \langle[0.6,0.7],[0.1,0.2]\rangle \\ c_5 & \langle[0.7,0.8],[0.1,0.2]\rangle & \langle[0.8,0.9],[0.0,0.1]\rangle & \langle[0.8,0.9],[0.0,0.1]\rangle & \langle[0.5,0.6],[0.3,0.4]\rangle \end{array}$$

where $C=\{c_1,\ldots,c_5\}$, $V=\{v_1,\ldots,v_4\}$ and for $1\leq j\leq 5, 1\leq e\leq 4$: pk_{c_j,v_e,h_f} are IVIFPs. We construct $B=R^T\odot_{(\circ,*)}PK$

$$= \begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_4 \\ \hline k_1 & \langle[0.82,0.95],[0,0.003]\rangle & \langle[0.81,0.94],[0,0.004]\rangle & \langle[0.77,0.92],[0,0.005]\rangle & \langle[0.76,0.91],[0.0004,0.01]\rangle \\ k_2 & \langle[0.87,0.96],[0,0.006]\rangle & \langle[0.89,0.97],[0,0.005]\rangle & \langle[0.81,0.93],[0,0.009]\rangle & \langle[0.82,0.94],[0.0002,0.02]\rangle \\ k_3 & \langle[0.86,0.97],[0,0.007]\rangle & \langle[0.87,0.97],[0,0.007]\rangle & \langle[0.81,0.95],[0,0.01]\rangle & \langle[0.81,0.95],[0.0003,0.02]\rangle \\ k_4 & \langle[0.89,0.98],[0,0.006]\rangle & \langle[0.90,0.98],[0,0.006]\rangle & \langle[0.82,0.95],[0.00004,0.01]\rangle & \langle[0.84,0.96],[0.0002,0.01]\rangle \end{array},$$

which contains the cumulative optimistic estimates of the k_i-th candidate (for $1\leq i\leq 4$) for the v_e-th vacancy (for $1\leq e\leq 4$).
Step 4. We apply the optimistic aggregation operation.
We can conclude that k_4 is the optimal outsourcing provider for all services, respectively: v_1 - with degree of acceptance (d.a.) $\in[0.89,0.98]$; v_2 - with d.a. $\in[0.9,0.98]$; v_3 - with d.a. $\in[0.82,0.95]$ and v_4 - with d.a. $\in[0.84,0.96]$.

After application of IVIF Hungarian algorithm [18], we find that k_1 is the optimal provider for service v_3, k_2 - for the service v_1, k_3 - for the service v_4 and k_4 – for the service v_2.
Step 5. After application of the interval-valued form of ICrA with $\alpha=[0.80;0.90]$ and $\beta=[0;0.10]$ over R we determine that there are not criteria in a consonance.
Step 6. If all experts' estimations are correct then we obtain their new ratings as follows: for the expert d_1 - $\{\langle[0.73;0.82][0;0.09]\rangle$, for the d_2 - $\{\langle[0.64;0.73][0;0.09]\rangle$ and for the d_3 - $\{\langle[0.82;0.91][0;0.09]\rangle$.

IV. Conclusion

We have applied our newly proposed IVIFIMOA algorithm on real data from an oil refinery and have shown how it can be used to select the most eligible candidates for outsourcing company services. The proposed algorithm can be easily generalized for multidimensional IF data [9] and can be applied to MCDM with both exact and IF parameters. For future work we will further develop the software utility and will analyse more real datasets.

References

[1] S. Tranev, *Outsourcing conflicts in the transport infrastructure of a company,* Dissertation. Prof. Dr. A. Zlatarov University, Burgas; 2012 (in Bulgarian).

[2] D. Mavrov, *Software Implementation and Applications of Index Matrices,* Dissertation. Prof. Dr. A. Zlatarov University, Burgas; 2016 (in Bulgarian).

[3] H. J. Zimmerman,, *Fuzzy sets, decision making, and expert systems,* Boston: Kluwer Academic Publishers; 1987.

[4] K. Atanassov, *Index Matrices: Towards an Augmented Matrix Calculus. Studies in Computational Intelligence*, Springer, Cham, vol. 573; DOI: 10.1007/978-3-319-10945-9, 2014.

[5] K. Atanassov, "On Intuitionistic Fuzzy Sets Theory," *STUDFUZZ,* vol. 283, Springer, Heidelberg, DOI: 10.1007/978-3-642-29127-2, 2012.

[6] K. Atanassov, "Interval-valued intuitionistic fuzzy sets," *Studies in Fuzziness and Soft Computing,* Springer, vol. 388, 2020.

[7] K. Atanassov, G. Gargov, "Interval valued intuitionistic fuzzy sets," *Fuzzy sets and systems,* vol. 31 (3), 1989, pp. 343-349.

[8] K. Atanassov, "Extended Interval Valued Intuitionistic Fuzzy Index Matrices," *In: Atanassov K. et al. (eds) Uncertainty and Imprecision in Decision Making and Decision Support: New Challenges, Solutions and Perspectives, IWIFSGN 2018,* Advances in Intelligent Systems and Computing, vol. 1081, Springer, Cham, 2020.

[9] K. Atanassov, "n-Dimensional extended index matrices Part 1," *Advanced Studies in Contemporary Mathematics*, vol. 28 (2), 2018, pp. 245-259.

[10] K. Atanassov, D. Mavrov, V. Atanassova, "Intercriteria decision making: a new approach for multicriteria decision making, based on index matrices and intuitionistic fuzzy sets," *Issues in IFSs and Generalized Nets,* vol. 11, 2014, pp. 1-8.

[11] K. Atanassov, E. Szmidt, J. Kacprzyk, V. Atanassova, "An approach to a constructive simplication of multiagent multicriteria decision making problems via ICrA," *Comptes rendus de lAcademie bulgare des Sciences,* vol. 70 (8), 2017, pp. 1147-1156.

[12] K. Atanassov, P. Vassilev, J. Kacprzyk, E. Szmidt, "On interval-valued intuitionistic fuzzy pairs," *Journal of Universal Mathematics,* vol. 1 (3), 2018, pp. 261-268.

[13] K. Atanassov, P. Marinov, V. Atanassova, "InterCriteria analysis with interval-valued intuitionistic fuzzy evaluations," *in: Cuzzocrea A., Greco S., Larsen H., Saccà D., Andreasen T., Christiansen H. (eds) Flexible Query Answering Systems, FQAS 2019,* Lecture Notes in Computer Science, Springer, Cham, vol. 11529, 2019, pp. 329-338.

[14] K. Atanassov, P. Vassilev, O. Roeva, "Level Operators over Intuitionistic Fuzzy Index Matrices," *Mathematics,* vol. 9, 2021, pp. 366.

[15] L. Zadeh, *Fuzzy Sets,* Information and Control, vol. 8 (3), 338-353; 1965.

[16] R. Yager, "Non-numeric multi-criteria multi-person decision making," *International Journal of Group Decision Making and Negotiation,* vol. 2, 1993, pp. 81–93.
method for logistics outsourcing provider selection," *Knowledge-Based Systems,* 2015.

[17] V. Atanassova, O. Roeva, "Computational complexity and influence of numerical precision on the results of intercriteria analysis in the decision making process," *Notes on Intuitionistic Fuzzy Sets,* vol. 24 (3), 2018, pp. 53-63.

[18] V. Traneva, S. Tranev, "An Interval-Valued Intuitionistic Fuzzy Approach to the Assignment Problem," *In: Kahraman C. et al. (eds) Intelligent and Fuzzy Techniques in Big Data Analytics and Decision Making, INFUS 2019,* Advances in Intelligent Systems and Computing, vol. 1029, Springer, Cham, 2020, pp. 1279-1287.

[19] V. Traneva, S. Tranev, "Intuitionistic Fuzzy Index-Matrix Selection for the Outsourcing Providers at a Refinery," *INFUS 2021,* Advances in Intelligent Systems and Computing, Springer, Cham, 2021.

[20] V. Traneva, S. Tranev, "Intuitionistic Fuzzy Approach for Outsourcing Provider Selection in a Refinery," *In: S. Margenov, I. Lirkov (eds.) Proceedings of LSSC 2021, Sozopol, Bulgaria,* Lecture Notes in Computer Science, Springer, Cham, 2021 (in press).

[21] V. Traneva, S. Tranev, V. Atanassova, " Three-Dimensional Interval-Valued Intuitionistic Fuzzy Appointment Model", *In: S. Fidanova (eds.) Recent Advances in Computational Optimization,* Studies in Computational Intelligence, vol. 838, Springer, Cham, 2020, pp. 181-199.

[22] V. Traneva, S. Tranev, M. Stoenchev, K. Atanassov, " Scaled aggregation operations over two- and three-dimensional index matrices," *Soft computing,* vol. 22, 2019, pp. 5115-5120.

[23] V. Traneva, S. Tranev, *Index Matrices as a Tool for Managerial Decision Making,* Publ. House of the USB; 2017 (in Bulgarian).

[24] V. Traneva, D. Mavrov, S. Tranev, "Fuzzy Two-Factor Analysis of COVID-19 Cases in Europe," *2020 IEEE 10th International Conference on Intelligent Systems, IS 2020 - Proceedings*, 2020, pp. 533–538.

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 255–258
DOI: 10.15439/2021F95
ISSN 2300-5963 ACSIS, Vol. 25

Optimization and Evaluation of Calibration for Low-cost Air Quality Sensors: Supervised and Unsupervised Machine Learning Models

Petar Zhivkov
Institute of Information and Communication Technologies -
Bulgarian Academy of Sciences (IICT-BAS)
acad. Georgi Bonchev str. Bl. 2, 1113, Sofia, Bulgaria
Email: pzhivkov@iit.bas.bg

***Abstract*—With the advancement of air pollution management, low-cost sensors are increasingly being used in air quality monitoring, but the data quality of these sensors is still a major source of concern. In this paper, data from five air monitoring stations in Sofia were compared to data from fixed low-cost PM sensors. The values of atmospheric pressure from low-cost sensors and the effects of relative humidity were investigated. A two-step model was created to refine the calibration process for low-cost PM sensors. At first, we calibrated the sensors with five separate supervised machine learning models and then the ANN-final model with anomaly detection completed the results. The ANN-final model improved the R^2 values of the PM10 determined by low-cost sensors from 0.62 to 0.95 as compared to standard instruments. In conclusion, the two-step calibration model proved to be a positive solution to addressing low-cost sensor efficiency issues.**

I. Introduction

AIR POLLUTION is a significant public health problem that has long been a source of anxiety for citizens.An air pollutant is described as any substance that can affect humans, animals, plants, or materials. In the case of humans, an air pollutant may cause or lead to an increase in mortality or serious illness, as well as pose a current or potential health risk [1]. Measurements of air emissions are critical for epidemiology and air quality control, but the scope of ground-based air pollution observations has limitations [2]. PM (Particulate Matter) air pollution is a suspended combination of solid and liquid particles that vary in quantity, size, shape, surface area, chemical composition, solubility, and origin. Total suspended particles (TSPs) have a trimodal size distribution in the ambient air, including coarse particles, fine particles, and ultrafine particles [3]. PM size-selective sampling refers to the collection of particles that are below, above, or within a defined aerodynamic range of sizes, which is commonly chosen to be relevant to inhalation and deposition, causes, or toxicity [4].

A. Air quality monitoring systems

Traditionally, concentrations of air emissions have been monitored by air monitoring stations equipped with standard equipment, allowing for highly reliable monitoring results. However, the high costs of equipment and servicing make meeting the demands of high-resolution surveillance and assessing the extent of personal exposure impossible [5], [6]. Low-cost air quality sensors have been widely used in air monitoring in recent years due to the benefits of low cost, low power usage, quick operation, and rapid response [7].

B. Opportunities and disadvantages of wireless low-cost stations

Wireless sensor networks connect a large number of fixed sensors in multiple places into a single network, enabling long-term, high-resolution surveillance of air contaminants [8]. Its applications are seen often in health-related studies and tracking individual exposure. A sensor tracking network was built on the Hong Kong marathon route in 2015, and it is used to measure the Air Quality Health Index (AQHI) and determine athletes' individual exposure levels [9]. In Rochester, NY, USA, a study used PM sensors to test airborne PM at various locations concurrently and continuously to determine the temporal and spatial variance of PM, as well as the impact of traffic and wood burning on outdoor PM concentrations [10]. By installing sensors on carriers such as vehicles, motorcycles, and drones, the mobile sensor network can provide more compact spatial data than fixed sensors and can achieve stereoscopic tracking of air quality and emission sources [11].

C. Effects from humidity and height

Despite the low-cost sensor's widespread usage, the data's precision has been challenged [12]. Some countries have conducted sensor assessment and calibration tests, and recommendations for the use and evaluation of sensors have been written [13]. However, just a few researchers have tested and calibrated low-cost sensors so far, and their performance under a variety of environmental conditions and time scales is still unknown [14]. Relative humidity (RH) and particle size distribution have been shown to have a significant effect on sensor monitoring results [15]. Other research investigated in foggy conditions at high RH (RH > 80%) the tracked performance of the sensor was higher than that of the normal instrument [16]. Several studies identify that barometric pressure is important for modeling particulate matter because complicated wind flows may occur, resulting in stagnant/stationary conditions with little circulation. Pollutants accumulate near the ground as a result of these conditions [17]. When barometric pressure was included in the model, the connection between particulate matter and cardiovascular mortality was marginally strengthened [18]. Finding associations between MLH and near-surface pollutant concentrations representative for a city like Berlin (flat terrain) appears to be impractical, particularly when traffic emissions are dominant [19].

II. Methodology

The efficiency of fixed low-cost sensors was evaluated in this study using five air quality monitoring stations in Sofia (capital city of Bulgaria) which were compared with standard instruments. The effects of AP, RH, and PM size distribution (PM10 ratio) on the performance of PM sensors were analyzed. Taking independent variables (RH, T and AP) and the dependent variable (PM10) as input factors, a two-step model calibration model to adjust fixed sensors was designed. Finally, we evaluated the performance of each model and gave recommendations for the conditions of the model application.

A. Reference instrument

This research used five air quality control stations with traditional measuring methods as a guide. The BETA-attenuation monitor (BAM) was used to calculate PM10. Impactors, cyclones, detection parts, and a dynamic heating system are among the standard instruments. They are situated in Sofia, Bulgaria respectively in the areas of Mladost, Druzhba, Nadezhda, Hipodruma, and Krasno Selo. Only one of these stations is measuring PM 2.5 and due to this PM 2.5 is not used for a reference in this research.

B. PM sensors

The PM sensor used in this research was the NovaFitness SDS011 laser particle sensor. The laser diffraction principle is used to use the sensor. The laser illuminates the trapped ions when capturing dispersed light waves at a certain angle as air flows through the photosensitive region of the sensor. A particle size continuum is created by classifying these pulse signals into various particle size intervals in order to measure the mass concentration of the particles [20].

C. The wireless network

In this research the Wireless Sensor Network (WSN) of Luftdaten was used. It was made up of 300 fixed sensors covering Sofia. Each sensor was installed in a plastic tube that could be mounted on walls, balconies, street light poles, and other structures.

Certain guidelines were developed to aid in obtaining the best representation of PM emissions in the city with the least amount of sensors possible. The WSN used fixed sensors that were mounted in 1 km grids to ensure that the majority of the downtown area was covered with adequate density.

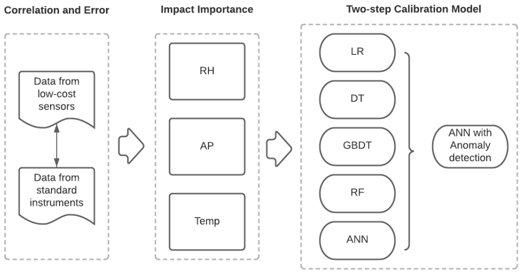

Correlation and Error
Impact Importance
Two-step Calibration Model
Data from low-cost sensors
Data from standard instruments
RH
AP
Temp
LR
DT
GBDT
RF
ANN
ANN with Anomaly detection

Fig. 1. Calibration Model

D. Calibration model setup and methods

The five sensors were situated right next to the five air monitoring stations. Since the sensor's time resolution differed from that of the regular instrument, the hourly mean was used for calculation and evaluation. A two-step model calibration method was modeled in this analysis, as seen in Fig. 1. Values of PM2.5 were measured only from one reference instrument, therefore, it was decided to exclude this instance from the model. A decision tree (DT) is a decision-making model that employs a tree-like model of decisions and their potential consequences, including the implications of chance events [21]. The Gradient Boosting Decision Tree (GBDT) algorithm is an iterative decision tree algorithm made up of multiple decision trees [22]. To obtain the final answer, the conclusions of all trees are added together. Random Forest (RF) is a tree predictor hybrid in which each tree is based on the values of a random variable sampled independently and with the same distribution for all trees in the forest [23]. In this research a RF with 10 trees is applied. The Artificial Neural Network (ANN) is a mathematical model that simulates neuronal behavior and is automatically modified by back-propagation errors [24]. Anomaly detection is decided It is a method of detecting unusual objects or occurrences in data sets that are out of the ordinary [25].The anomaly detector was used to remove outliers from the training dataset. As this is unsupervised learning, an evaluation with the same ANN setup was made, before and after cleaning the dataset to identify if unsupervised learning is appropriate for this dataset. RF and ANN are stochastic techniques, therefore, several runs have been performed in order to obtain objective results. In addition, statistical test from multiple runs have shown if the difference between the standard ANN and the one with anomaly detection is indeed significant.

The learning process of the model was divided into two stages: learning and testing. The raw data was split into two data sets at random, with 80% for training and 20% for testing. The model was first trained using the training data, and then its output was evaluated by the test data set.

E. Correlational analysis for atmospheric pressure

For evaluating the pressure measurement from the low-cost sensor were used the values from the reference instrument, the height difference between each sensor and station, and the Barometric formula. To do this 4 low-cost sensors within a perimeter of 500m on different heights were evaluated. The Barometric formula (1) is used to model how the pressure of the air changes with altitude and it is as follows:

$$P = P_b. \left[\frac{T_b + L_b.(h - h_b)}{T_b}\right]^{\frac{-g_0.M}{R^*.L_b}} \tag{1}$$

To evaluate the AP results from low-cost sensors a correlation method was used. The comparison statistical methods fall into two categorizations: parametric and nonparametric. Parametric comparisons are based on the premise that the variable is continuous and normally distributed. Nonparametric approaches are used where data is continuous with non-normal distribution or any other form of data other than continuous variables. As our calculation model includes data normality and strict sample size, a parametric method is chosen. Moreover, parametric methods are better ways to measure the difference between the groups relative to their equivalent nonparametric methods.

The parametric Pearson correlation coeficient (2) is used for comparing the two sources of data. It provides a measure of the linear association between the two continuous variables (usually just referred to as correlation coefficient). Correlation coefficients for each (x , y) pair are determined to carry out the evaluation, and the values of x and y are consequently replaced by their ranks. Applying the test findings to a coefficient of correlation ranging from -1 to 1.

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{2}$$

III. Results and Discussion

A. Comparison between low-cost sensors and standard instruments

For better evaluation of the low-cost sensors together with the coefficient of determination (R squared), the mean absolute error and mean squared error are calculated (see Table I). The mean value of R squared between PM sensors and standard instruments without calibration was 0.62. The LR model showed worst result with a mean R^2 of 0.77. The best correlation for PM10 came from the ANN model. The mean value of the R^2 was 0.94 (PM10), which matched the findings of previous studies [12], [2]. For long-term comparison, the low-cost sensor and the regular instrument were put in the same location, which was a common approach for sensor evaluation in previous studies [26].

B. Relative humidity and air pressure

Relative humidity (RH) and temperature are considered to be the most important impact factors on particle sensor efficiency. High RH has been shown in previous studies to be the catalyst to causing hygroscopic growth of particles and modifying optical properties, resulting in substantial interference for PM sensors [27]. Moreover, RH turned out to be of the highest importance in the RF and ANN models for the PM10 values.

The low-cost sensors had identical values as compared to normal instruments when RH was below 40%. While PM10 had a poor correlation when RH was above 80%.

Results of the AP from 4 sensors installed on different heights within 500m were compared with the calculations from the barometric formula and the data from the reference instrument. The installation height of the sensors was 3, 6, 8, and 18m while the height of the reference instrument is 2m. Calculations showed a high correlation between the sensors and r the with a mean value of the Pearson correlation coefficient r = 0,92.

C. Results of the calibration model

Table I presents the statistical outcomes of each model's testing, where Mean Absolute Error (MAE), Mean Squared Error (MSE), and the R^2 were determined.

The output of the other five separate models showed that the ANN model performed best. The RF model showed slightly worse results. The R^2 of PM10 increased from 0.62 to 0.9 and 0.94 for RF and ANN respectively. The ANN model performed best out of the 5 models, slightly better than the RF model, therefore, was chosen to be used in comparison with anomaly detection.

D. Improvements of the model through unsupervised learning

The ANN model was used as an autonomous model with sensor data and environmental variables as inputs. With the output values of five independent models as inputs, the ANN-final model conducted an artificial neural network model after filtering the dataset from anomalies . It was set up to find the top 20 anomalous instances within a forest with 256 trees. The highest anomaly score of a tree was 68%. These 20 outliers were removed from the training set and ran an ANN with the cleaned model which was evaluated again. The compared evaluations showed improvement with an R^2 of 0.95. In addition, the MAE and MSE decreased by 5.16% and 14.69% respectively. Therefore, the use of unsupervised learning in this study is considered to be useful. In conclusion, the ANN-final model had the best calibration score, with the largest R squared and the best correlation, indicating that the two-step model was more accurate than a single model in the model calibration of low-cost sensors.

IV. Conclusion and future research

The efficiency of PM sensors was measured by comparing by standard instruments using the wireless sensor network. To calibrate the fixed sensors, a two-step process was developed, and the model's results were evaluated. The following are the major conclusions: The findings of the two-step model were satisfactory. The R^2 of the fixed PM10 sensors increased from 0.62 to 0.95. The ANN model had the strongest impact of the five independent models, followed by the RF model,

TABLE I
RESULTS FROM SUPERVISED LEARNING MODELS.

TYPE OF MODEL	MEAN ABSOLUTE ERROR	MEAN SQUARED ERROR	R SQUARED
LINEAR REGRESSION:	11.19	288.12	0.77
DECISION TREE:	8.89	170.03	0.86
GRADIENT BOOSTING DECISION TREE:	8.68	145.22	0.89
RANDOM FOREST:	7.96	125.57	0.90
ARTIFICIAL NEURAL NETWORK:	6.27	83.90	0.94

while the LR model was ineffective. Anomaly detectors can be an unsupervised alternative to classifiers in an unbalanced dataset and in this research the final result was improved. The atmospheric pressure values of 4 low cost sensors were compared with a standard station by the use of calculations with the Barometric formula. The correlation was strong which means that low-cost sensors may be considered as a good source of modeling air pollution in vertical planning in further researches.

Further studies will beneficial in incorporating gas sensors into the WSN network. In addition, it is useful to analyze automotive emissions with integrated mobile sensors in the vehicles and using the model's improvement from this research.

REFERENCES

[1] M. Kampa and E. Castanas, "Human health effects of air pollution," *Environmental pollution*, vol. 151, no. 2, pp. 362–367, 2008.

[2] X. Qin, L. Hou, J. Gao, and S. Si, "The evaluation and optimization of calibration methods for low-cost particulate matter sensors: Inter-comparison between fixed and mobile methods," *Science of The Total Environment*, vol. 715, p. 136791, 2020.

[3] C. A. Pope III and D. W. Dockery, "Health effects of fine particulate air pollution: lines that connect," *Journal of the air & waste management association*, vol. 56, no. 6, pp. 709–742, 2006.

[4] J. C. Chow, "Measurement methods to determine compliance with ambient air quality standards for suspended particles," *Journal of the Air & Waste Management Association*, vol. 45, no. 5, pp. 320–382, 1995.

[5] P. Mouzourides, P. Kumar, and M. K.-A. Neophytou, "Assessment of long-term measurements of particulate matter and gaseous pollutants in south-east mediterranean," *Atmospheric Environment*, vol. 107, pp. 148–165, 2015.

[6] J. Y. Chin, T. Steinle, T. Wehlus, D. Dregely, T. Weiss, V. I. Belotelov, B. Stritzker, and H. Giessen, "Nonreciprocal plasmonics enables giant enhancement of thin-film faraday rotation," *Nature communications*, vol. 4, no. 1, pp. 1–6, 2013.

[7] Y. Wang, J. Li, H. Jing, Q. Zhang, J. Jiang, and P. Biswas, "Laboratory evaluation and calibration of three low-cost particle sensors for particulate matter measurement," *Aerosol Science and Technology*, vol. 49, no. 11, pp. 1063–1077, 2015.

[8] A. R. Rasyid, N. P. Bhandary, and R. Yatabe, "Performance of frequency ratio and logistic regression model in creating gis based landslides susceptibility map at lompobattang mountain, indonesia," *Geoenvironmental Disasters*, vol. 3, no. 1, pp. 1–16, 2016.

[9] L. Sun, J. Wei, D. Duan, Y. Guo, D. Yang, C. Jia, and X. Mi, "Impact of land-use and land-cover change on urban air quality in representative cities of china," *Journal of Atmospheric and Solar-Terrestrial Physics*, vol. 142, pp. 43–54, 2016.

[10] N. Zikova, M. Masiol, D. C. Chalupa, D. Q. Rich, A. R. Ferro, and P. K. Hopke, "Estimating hourly concentrations of pm2. 5 across a metropolitan area using low-cost particle monitors," *Sensors*, vol. 17, no. 8, p. 1922, 2017.

[11] M. Mead, O. Popoola, G. Stewart, P. Landshoff, M. Calleja, M. Hayes, J. Baldovi, M. McLeod, T. Hodgson, J. Dicks *et al.*, "The use of electrochemical sensors for monitoring urban air quality in low-cost, high-density networks," *Atmospheric Environment*, vol. 70, pp. 186–203, 2013.

[12] A. C. Rai, P. Kumar, F. Pilla, A. N. Skouloudis, S. Di Sabatino, C. Ratti, A. Yasar, and D. Rickerby, "End-user perspective of low-cost sensors for outdoor air pollution monitoring," *Science of The Total Environment*, vol. 607, pp. 691–705, 2017.

[13] H. Brantley, G. Hagler, E. Kimbrough, R. Williams, S. Mukerjee, and L. Neas, "Mobile air monitoring data-processing strategies and effects on spatial air pollution trends," *Atmospheric measurement techniques*, vol. 7, no. 7, pp. 2169–2183, 2014.

[14] B. Zheng, D. Tong, M. Li, F. Liu, C. Hong, G. Geng, H. Li, X. Li, L. Peng, J. Qi *et al.*, "Trends in china's anthropogenic emissions since 2010 as the consequence of clean air actions," *Atmospheric Chemistry and Physics*, vol. 18, no. 19, pp. 14 095–14 111, 2018.

[15] A. Mukherjee and M. Agrawal, "World air particulate matter: sources, distribution and health effects," *Environmental Chemistry Letters*, vol. 15, no. 2, pp. 283–309, 2017.

[16] R. Jayaratne, X. Liu, P. Thai, M. Dunbabin, and L. Morawska, "The influence of humidity on the performance of a low-cost air particle mass sensor and the effect of atmospheric fog," *Atmospheric Measurement Techniques*, vol. 11, no. 8, pp. 4883–4890, 2018.

[17] B. Murthy, R. Latha, A. Tiwari, A. Rathod, S. Singh, and G. Beig, "Impact of mixing layer height on air quality in winter," *Journal of Atmospheric and Solar-Terrestrial Physics*, vol. 197, p. 105157, 2020.

[18] N. Janssen, P. Fischer, M. Marra, C. Ameling, and F. Cassee, "Short-term effects of pm2. 5, pm10 and pm2. 5–10 on daily mortality in the netherlands," *Science of the Total Environment*, vol. 463, pp. 20–26, 2013.

[19] A. Geiß, M. Wiegner, B. Bonn, K. Schäfer, R. Forkel, E. v. Schneidemesser, C. Münkel, K. L. Chan, and R. Nothard, "Mixing layer height as an indicator for urban air quality?" *Atmospheric Measurement Techniques*, vol. 10, no. 8, pp. 2969–2988, 2017.

[20] K. A. Koehler and T. M. Peters, "New methods for personal exposure monitoring for airborne particles," *Current environmental health reports*, vol. 2, no. 4, pp. 399–411, 2015.

[21] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE transactions on systems, man, and cybernetics*, vol. 21, no. 3, pp. 660–674, 1991.

[22] S. W. Kwok and C. Carter, "Multiple decision trees," in *Machine Intelligence and Pattern Recognition*. Elsevier, 1990, vol. 9, pp. 327–335.

[23] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[24] M. Gevrey, I. Dimopoulos, and S. Lek, "Review and comparison of methods to study the contribution of variables in artificial neural network models," *Ecological modelling*, vol. 160, no. 3, pp. 249–264, 2003.

[25] D. Savage, X. Zhang, X. Yu, P. Chou, and Q. Wang, "Anomaly detection in online social networks," *Social Networks*, vol. 39, pp. 62–70, 2014.

[26] D. H. Hagan, G. Isaacman-VanWertz, J. P. Franklin, L. M. Wallace, B. D. Kocar, C. L. Heald, and J. H. Kroll, "Calibration and assessment of electrochemical air quality sensors by co-location with regulatory-grade instruments," *Atmospheric Measurement Techniques*, vol. 11, no. 1, pp. 315–328, 2018.

[27] X. Liu, R. Jayaratne, P. Thai, T. Kuhn, I. Zing, B. Christensen, R. Lamont, M. Dunbabin, S. Zhu, J. Gao *et al.*, "Low-cost sensors as an alternative for long-term air quality monitoring," *Environmental research*, vol. 185, p. 109438, 2020.

Advances in Computer Science and Systems

ACSS is welcoming presentations of the scientific aspects related to applied sciences. The session is oriented on the research where the computer science meets the real world problems, real constraints, model objectives, etc. However the scope is not limited to applications, we all know that all of them were born from the innovative theory developed in laboratory. We want to show the fusion of these two worlds. Therefore one of the goals for the session is to show how the idea is transformed into application, since the history of modern science show that most of successful research experiments had their continuation in real world. ACSS session is going to give an international panel where researchers will have a chance to promote their recent advances in applied computer science both from theoretical and practical side.

Scope:

- Applied Artificial Intelligence
- Applied Parallel Computing
- Applied methods of multimodal, constrained and heuristic optimization
- Applied computer systems in technology, medicine, ecology, environment, economy, etc.
- Theoretical models of the above computer sciences developed into the practical use

Track Chairs

- **Dimov, Ivan,** Bulgarian Academy of Sciences, Institute of Information and Communication Technologies, Bulgaria
- **Wasielewska-Michniewska, Katarzyna,** Systems Research Institute, Polish Academy of Sciences, Poland

Program Chairs

- **Dimov, Ivan,** Bulgarian Academy of Sciences, Institute of Information and Communication Technologies, Bulgaria
- **Wasielewska-Michniewska. Katarzyna,** Systems Research Institute, Polish Academy of Sciences, Poland

Program Committee

- **Barbosa, Jorge,** University of Porto, Portugal
- **Braubach, Lars,** University of Hamburg, Germany
- **Cabri, Giacomo,** Università di Modena e Reggio Emilia, Italy
- **Fabijańska, Anna,** Technical University of Lodz, Poland
- **Georgiev, Krassimir,** Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, Bulgaria
- **Homles, Violeta,** University of Huddersfield, United Kingdom
- **Jezic, Gordan,** University of Zagreb, Croatia
- **Kotenko, Igor,** St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS), Russia
- **Lirkov, Ivan,** Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, Bulgaria
- **Mangioni, Giuseppe,** Dipartimento di Ingegneria Elettrica Elettronica e Informatica (DIEEI) - University of Catania, Italy
- **Millham, Richard,** Durban University of Technology, South Africa
- **Modoni, Gianfranco,** STIIMA-CNR, Italy
- **Pandey, Rajiv,** Amity University, India
- **Pawłowski, Wiesław,** University of Gdańsk, Poland
- **Petcu, Dana,** West University of Timisoara, Romania
- **Scherer, Rafał,** Częstochowa University of Technology, Poland
- **Schreiner, Wolfgang,** Research Institute for Symbolic Computation (RISC), Austria
- **Tudoroiu, Nicolae,** John Abbott College, Canada
- **Wyrzykowski, Roman,** Częstochowa University of Technology, Poland
- **Vardanega, Tullio,** University of Padua, Italy

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 261–269
DOI: 10.15439/2021F34
ISSN 2300-5963 ACSIS, Vol. 25

Thespis: Causally-consistent OLTP

Carl Camilleri, Joseph G. Vella, Vitezslav Nezval
Computer Information Systems
University of Malta, Malta
Email: {carl.camilleri.04, joseph.g.vella, vitezslav.nezval}@um.edu.mt

***Abstract*—Data Consistency defines the validity of a data set according to some set of rules, and different levels of data consistency have been proposed. Causal consistency is the strongest type of consistency possible when data is stored in multiple locations, and fault tolerance is desired. Thespis is a middleware that leverages the Actor model to implement causal consistency over a DBMS, whilst abstracting complexities for application developers behind a REST interface. ThespisTRX is an extension that provides read-only transaction capabilities, whilst ThespisDIIP is another extension that handles distributed integrity invariant preservation. Here, we analyse standard transactional workloads on the relational data model, which is richer than the key-value data model supported by the Thespis interface. We show the applicability of the Thespis approach for this data model by designing new operations for the Thespis interface, which ensure correct execution of such workloads in a convergent, causally-consistent distributed environment.**

I. Introduction

THE CAP theorem [1], [2] proves that having both availability and partition tolerance within dispartite databases (DBs) that implement Strong Consistency (SC) [3] is not possible. SC is the strongest type of consistency offered by traditional distributed database management systems (DDBMSs), that manage databases across multiple nodes.

Distributed data centres have led to a wide adoption of DBs that forego strong data consistency in favour of availability and partition tolerance to provide the scalability and high availability properties sought by enterprise-scale applications. Popular DBs in this area offer Eventual Consistency (EC) [4], a weak consistency model which guarantees that given no new WRITE operations, all nodes (i.e. distributed partitions) of the DB eventually converge to the same state.

EC is relatively easy to achieve, and does not suffer from the performance limitations of distributed algorithms, such as Paxos [5], that attempt to achieve a degree of availability and SC in a distributed environment. However, EC shifts data safety and consistency responsibilities to the application layer, giving rise to a new set of problems [6].

Causal Consistency (CC) [7] is weaker than SC, but stronger than EC, and has been proven to be the strongest type of consistency that can be achieved in a fault-tolerant, distributed system [8]. Informally, CC implies that readers cannot find a version of a data element before all the operations that led to that version are visible [9].

II. Problem Domain

CC is sufficiently strong, and sufficiently performant, for most enterprise applications [10]. However, we believe its adoption in the industry is compounded by a number of aspects, including:

1) Lack of support for rich data modeling, with CC DBs supporting data sets based on the key-value data model, or abstract data types.
2) Programmer accessibility, given that existing CC DBs require engineering applications specifically around their semantics or client libraries, and do not sufficiently abstract the programmer from the complexities of distributed data management [11].
3) Database lock-in, with the CC DB storing data in native formats which are incompatible with other consumers.

We propose a middleware that achieves CC, stores the data in a relational database management system (RDBMS), and integrates with applications through intuitive APIs, abstracting the complexities of CC as much as possible. In this paper, we also analyse online transactional processing (OLTP) workloads that are traditionally used to benchmark widely-adopted RDBMSs, and propose the semantics that such a middleware should offer to enable an application to perform the same function within a CC DDBMS.

III. Definitions

This section identifies some terminology and provides relevant definitions, as used throughout the rest of the paper.

Replica. A replica denotes a copy of a database. Our context assumes that each replica is a full copy of the database.

Data Centre/Node. A Data Centre (DC), or node, refers to a physical location that hosts one of the replicas of a database.

Distributed Database. A distributed database (DDB) is considered to be a database which resides in multiple nodes.

Database Operation. A database operation denotes an activity that is performed by an application against some API offered by the database, or by the overlying middleware.

Operation Latency. The operation latency is the time elapsed between when a client submits a database operation to when the result of that operation arrives back at the client (typically quoted in milliseconds).

System Throughput. The throughput achieved from a particular setup (i.e. system implementation installed on a specific infrastructure) is the number of operations served in a period of time (typically quoted in requests per second).

Data Freshness. Data freshness refers to how long it takes clients connected to a remote DC to be able to read the result of a DB operation that changes state (typically, a WRITE operation) in the local DC. In most implementations, data freshness is traded for throughput and operation latency optimisation i.e. a higher throughput and lower latency are preferred over reading the latest changes from remote DCs.

Causal Consistency. A system is causally-consistent if all operations that are causally-related are seen in the same order across all the nodes in the DDB [7]. Two operations a and b are deemed to be potentially causally-related, denoted by $a \to b$ (i.e. a leads to b), if at least one of three criteria holds [12]:

1) **Thread of Execution:** Given a process P, operations within the same process are causally-related i.e. $a \to b$ if P performs a, then it performs b.
2) **Reads from:** Given a WRITE operation a, if b reads the result of a, then $a \to b$.
3) **Transitivity:** If $a \to b$, and $b \to c$, then $a \to c$.

Conversely, the order of *concurrent* operations across the different nodes is not guaranteed. Concurrent operations are those that are not related through causality, and are therefore essentially unrelated with respect to the data items read and/or written [7]. Two operations a and b are deemed to be concurrent if $a \nrightarrow b$ and $b \nrightarrow a$, and so can be replicated in any order across the DDB cluster, without violating CC.

Conflict Handling. In order for a DDB to support high availability with low latency, WRITEs need to be accepted at any node without requiring co-ordination, at least in the critical path, with other nodes [9]. A state of conflict causes consistency between different nodes to be broken. A conflict is declared when the same data element in two non-colocated replica gets updated concurrently. Two operations on the same data element are conflicting if they write a different value, and are not related by causality [7] i.e. they are concurrent. Concurrent and conflicting operations, in the context of a DDB, can therefore be defined as follows:

- Let Θ_1 and Θ_2 define a WRITE operation on a data item identified by key k, in DC_1 and DC_2 respectively.
- Let $\Theta_1 = put(k, v_1)$.
- Let $\Theta_2 = put(k, v_2)$.

$\therefore$ Θ_1 and Θ_2 are concurrent and conflicting operations.

Various approaches for conflict detection and conflict resolution have been put forward [13], [14], [15]. The Last-Writer Wins (LWW) is a popular conflict resolution technique where the most recent update is retained in case of conflict. A database which offers CC as well as conflict detection and resolution, and therefore convergence, is said to provide *causal+ consistency* (CC+) [16].

IV. Literature Review

A. Causally-Consistent Databases

In COPS [16], application clients are co-located with a cluster of servers that store a full replica of the DB. A WRITE operation is placed in a queue and sent to peer DCs, where it is stored if all its dependencies are also stored. Dependencies of a WRITE operation are tracked by a client-side library that tracks a *context identifier*. Within a *context*, dependencies of a WRITE operation are defined as the latest version of all keys that have so far been interacted with, guaranteeing causality. Conflicts are handled using a LWW approach.

COPS-GT [16] extends COPS with support for read-only transactions. Clients can request the values of a set of keys, rather than that of a single key, and the DDBMS returns a causally-consistent snapshot of the requested keys. COPS-GT, like COPS, uses a sequentially-consistent key-value store, but changes the client library, the DB and the semantics of the READ and WRITE operations. Keys are mapped to a set of versions for that key, rather than one value as in COPS. Each version is mapped to a value and a set of dependencies, encoded as pairs of <*k*ey,version>, thus supporting READ and WRITE operations in a transactional context.

Bolt-On [9] describes a custom middleware on top of Cassandra, a commercially-available EC DB with a columnar data model which handles replication. Bolt-On implements explicit causality, offloading dependency tracking to the client. Data items are tagged with a set of tuples, each indicating a process identifier and its monotonically-increasing identifier. The dependencies of a WRITE operation are the versions of the keys read to produce that operation. Each client holds an *interest set*, the keys that it needs to read, and a resolver process maintains a causally-consistent view of the keys within this set by fetching the latest versions of the data items and their dependencies.

GentleRain [17] provides CC over a key-value, multi-versioned, sharded and replicated DB. It depends on a custom replication protocol to propagate WRITEs across DCs. Dependency tracking is efficient, as the only meta-data stored with a WRITE operation is a timestamp and a server identifier. Any READ operation can only access versions of data that are created in the local DC, or versions that have been created in remote DCs and replicated across all DCs. This guarantees causality by ensuring that when reading a version, the items which have led to its creation (i.e. its dependencies) are present in all DCs.

Wren [18] takes a somewhat similar approach to [17], but uses Hybrid Logical Clocks (HLC) [19] to timestamp events in a more reliable manner. Furthermore, Wren implements transactional CC, allowing clients to perform read transactions as well as running multiple WRITE operations atomically.

B. Benchmark Workloads

DBMS query workloads are segmented into two broad modes [20], [21]. Online transactional processing (OLTP) workloads consist of WRITE queries that modify small amounts of data, and READ queries that process a few records and project the majority of the attributes available [22]. In OLTP, short response times are crucial to avoid user frustration and business impact [23]. In contrast, Online analytical processing (OLAP) workloads typically consist of read-only queries that traverse a large amount of records, performing aggregations and projecting a small set of attributes [22].

TABLE I
TPC-C TRANSACTIONS

Transaction	Characteristic	Minimum Percentage of mix
New Order	read-write	45%
Payment	read-write	43%
Order Status	read-only	4%
Delivery	read-write	4%
Stock Level	read-only	4%

The TPC-C [24] workload simulates a DB which models a number of geographically-distributed brick and mortar warehouses, each associated to one or more districts. A number of terminals perform transactions on stock available in each warehouse. A TPC-C workload is characterised by five transactions over nine tables containing synthetic data [25], as summarised in Table I. The benchmark specification also defines that an execution should comprise transactions chosen at random, but the final transaction set should maintain a minimum percentage for each type of transaction. Each transaction consists of a number of queries, as shown in Table II [26]. The workload can be throttled by two parameters. The scale factor (sf) specifies the number of records that are generated within the DB. The scale factor determines the number of warehouses available, which in turn determines the number of records generated in the other tables. Conversely, the number of terminals determines the number of parallel threads that the workload generator spawns to execute concurrent transactions. Hence, larger scale factors imply that TPC-C queries are heavier (e.g. record selections operators applied to larger tables), but possibly less contentious (a larger number of records reduces the probability of contention), whilst a larger number of terminals increases the number of concurrent transactions, and thus the probability of contention. Finally, TPC-C defines a set of tests that should be executed to confirm that the system under test guarantees suitable Atomicity, Consistency, Isolation, and Durability (ACID) properties.

The TPC-E [27] workload simulates a DB used by a financial brokerage firm that stores customer-related information (e.g. accounts, holdings, watch lists), broker information (e.g., trades, trade history), and financial market data (e.g., companies, securities, related news items, last trades). The TPC-E data model consists of 33 tables and twice the number of columns when compared to TPC-C, and is seeded with pseudo-real data based on the U.S. and Canada census from 2000, as well as census data and actual listings on the NYSE and NASDAQ stock exchanges [28]

The Smallbank [29] benchmark simulates a banking application that sustains transactions relative to financial accounts. The access pattern of the workload is skewed to define a small number of "hot" accounts upon which most transactions are executed. By nature of the application, transactions in this benchmark involve small number of records.

V. MIDDLEWARE FOR CAUSAL CONSISTENCY

As shown by our review of the literature, current approaches to CC offer very specific interfaces, depend on custom DBs, and offload data-handling functionality to client applications in a way that makes CC non-trivial to implement and use. The majority of the approaches also deal with a simple key-value data model. Although this data model can be the building block of more complex schemas, application developers are used to richer data model structures.

Thespis [30] is a middleware that provides CC as well as conflict detection and resolution, thus achieving CC+. Data is stored in a database which offers SC and a rich data model with effective querying and reporting capabilities on data generated by OLTP systems, but no support for horizontal scalability (i.e. the underlying database does not offer the possibility to scale out on more than one node out of the box). This fits the description of a RDBMS. Relational databases are widely used in production systems, and offer a richer data model with effective querying and reporting capabilities on data generated by OLTP systems. Hence, although not mandatory for the Thespis approach, the implementation assumes that: a) the main data backing engine is a RDBMS; and b) that the client is an application handling “objects”, primarily instances of business-domain models.

Thespis tackles several objectives: it enables the use of CC without requiring major application re-engineering, stores data in a format accessible to other systems that need to consume it (e.g. reporting modules), and considers efficiency such that performance overheads of CC guarantees do not outweigh the benefits of using a DDBMS.

These objectives are tackled through the fusion of a number of concepts, most importantly:

1) **The Actor Model** [31], which organises logic in terms of a hierarchical society of “experts” that communicate together via asynchronous message passing. An actor consists of a) a *mailbox* where incoming messages are queued; b) an actor’s *behaviour*, or the behavioural logic that is executed in response to a received message; and c) an actor’s *state*, in other words the data stored by the actor at a given point in time. Actors process one message at a time, and exist in the context of Actor Systems [32], where hierarchies can form.
2) **Command Query Responsibility Segregation (CQRS)** [33], a software design pattern that applies the concept of Command Query Separation (CQS) [34] in order to maintain separate data models for READs and for WRITEs.
3) **Event Sourcing (ES)** [35], another pattern where all data changes are captured as a sequence of events that are stored in an event log and that, when applied in order, provide a view of the system state at a particular point in time.

Figure 1 illustrates the Thespis middleware that offers an API allowing two operations, READ and WRITE. All operations employ the Actor model to deal with both concurrency issues.

Firstly, the actor-based implementation ascertains that READs happen concurrently. Secondly, it also ascertains that WRITEs on the same object, and in the same replica, happen in a set sequence. The hierarchical nature of Actor Systems is also exploited to reflect a causally-consistent view of the underlying database. The **Writer Actor** and **Reader Actor** are responsible for storing actor states and retrieving business objects from the underlying DB respectively. The **Replication Actor** is responsible for replicating actor state changes from one replica to the other. The core **Middleware** actor system holds a set of actors which provide a view of the underlying DB to the application. Finally the actor system adopts a "child-per-entity" approach, spawning one Entity Actor per type of business object (e.g. DB table), supervising dedicated Entity Instance Actors for each business object instance. The state of the Entity Instance Actor is made up of two elements: the Entity Instance and the Event Log. The events in the Event Log can be applied to the Entity Instance governed by the Entity Instance Actor to retrieve the latest (causally-consistent) version of the entity.

The middleware snapshots data changes in the DB only when received in all the DCs. WRITEs are captured in the middleware layer and, given a new version of an entity being created by any WRITE operation, a set of events representing the new state, compared to the previous version, are extracted.

Finally, the system incorporates a replication protocol, again founded on the Actor model, which encapsulates two algorithms, one running on the *Originating Server* (i.e. the server where a new event is created), the other on the *Remote Server*, or the server which is receiving an event from an *Originating Server*. Key to the replication protocol, and to enforce causality, is the Stable Version Vector (SVV). The SVV is simply a vector of length M, where M is the number of peer DCs. Each element SVV_{DC} in the vector is the latest observed timestamp from the corresponding peer DC. Specifically, the vector element $SVV_{DC_N}[M]$ denotes the latest timestamp observed from DC M within DC N.

Details of the implementation, performance evaluation and correctness assessment of the Thespis middleware are given in our previous work [30]. Results show that the Thespis approach achieves CC+, availability and partition tolerance. Furthermore, inline with the PACELC theorem [36], Thespis can provide CC, whilst optimising operation latency under normal conditions, as well as tolerating network partitions or node failures.

A. Read-only Transactions

ThespisTRX [37] adds to Thespis the functionality for when multiple entities need to be retrieved from the underlying DB, potentially in multiple operations whilst preserving causality. Our example in the context of a social media application [37] shows a common encounter of Time-To-Check-Time-To-Use (TOCTOU) race conditions using the Thespis API, and underlines the need for this extension.

ThespisTRX builds on the Thespis approach with three main additions. First of all, two new system components are introduced, namely:

1) The **Transaction Coordinator**, which is responsible to track transactions that are running in the local node at any point in time;
2) The **Entity Version Log**, which is responsible to hold versions of entities that may be required by all currently-running transactions, and which can easily be queried.

Both new components are implemented within the middleware, of which an instance exists in each DC.

Secondly, the middleware API is extended to support three new operations. STARTTRAN and ENDTRAN signal the start and end of a read transaction respectively, whilst READTRX is essentially an overload of the standard READ operation that takes an additional parameter *TransactionId.* Thirdly, the logic in the business application is slightly adjusted to signal the start and end of a transaction, and passes the transaction identifier to all read operations.

Thirdly, given this model, the logic in the business application is slightly adjusted too: the business application signals the start and end of a transaction, and adds the transaction identifier to all of its read operations.

Details of the implementation, performance evaluation and correctness assessment of the ThespisTRX extension are discussed in [37]. The results from empirical evaluation show that read latency is similar in both Thespis and ThespisTRX, confirming that READs do not interfere with the WRITEs in ThespisTRX that maintain the Entity Version Log. WRITEs have a slightly higher latency in ThespisTRX, which is expected due to the need to store additional meta data (i.e. Entity Version Log) in order to support transactional reads. Nonetheless, the reduction in throughput varies between 1.8% (for read-heavy workloads) and 4.8% (for write-heavy workloads), and therefore does not prohibit the use of ThespisTRX for the same workloads as Thespis.

B. Distributed Integrity Invariant Preservation

Integrity Invariants are application-specific operation pre-conditions, or rules that determine whether an operation on a data element should be accepted or not. A DDBMS such as Thespis [30] achieves low latency and high availability by allowing operations to be accepted at any DC, and propagated to other DCs asynchronously. The result of an operation accepted at any replica can be propagated to a remote replica at a time when the operation's pre-condition no longer holds [38], leading to an anomaly in the integrity invariant.

ThespisDIIP [39] is an extension of Thespis that brings distributed integrity invariant preservation (DIIP), over and above the original CC+ guarantees. Focus is given to integrity invariants for data values that must be satisfied according to a Linear Arithmetic Inequality (LAI) constraint [40]. These are a set of problems that involve resource allocation [41], such as operations on bank accounts (integrity invariants define that withdrawals cannot request more than the available funds) and order fulfillment operations (an order can be accepted only if there is enough stock). Although important in real-world applications [42], these types of integrity invariants are not *I-confluent* [43], meaning they cannot be preserved by concurrent transactions without co-ordination.

ThespisDIIP employs data-value partitioning (DVP) [44] and takes a novel approach to achieve DIIP by exploiting the

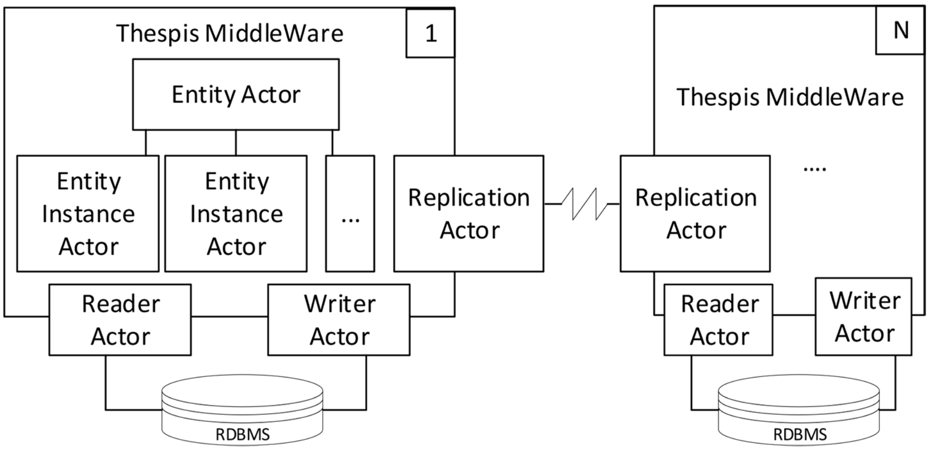


Fig. 1. Thespis Middleware System Model

structure of the underlying RDBMS to trigger background DVP operations when needed.

Design, implementation and benchmarking details are given in [39], where it is shown that the asynchronous operation latency in ThespisDIIP is short enough to eliminate waiting time in the critical path of a typical enterprise application.

VI. Applications to OLTP Workloads

The API offered by the Thespis middleware, as well as its extensions, is well-suited towards atomic workloads, such as YCSB [45], which is used to benchmark Thespis [30] and its extensions [37], [39], but more sophisticated OLTP workloads require richer semantics.

From the list of benchmarks described in our review of the literature, TPC-C is the oldest specification of a transactional workload, however it is also one that is established, and referred to in recent related and orthogonal works in the literature [46], [47]. We thus choose to focus on the TPC-C benchmark in this paper. First, we analyse its queries and subsequently design the necessary semantics to handle them through extensions to Thespis.

Table II summarises the TPC-C transactions and queries. Each of the latter is marked according to a relevant cluster for which semantics are proposed.

A. Primary Key Selection Operations

Queries that return zero or one records based on a predicate on the relation's primary key are trivial and can be satisfied by the Thespis API [30], or any CC DB using a key-value data model. These are marked with † in Table II.

B. Set Selection Operations

Marked with § in Table II, these are queries where the predicate is not based on a primary key, and that thus can return 0 or more records. This is considered feasible to achieve in the Thespis approach, whilst still preserving CC, as shown in Algorithm 1. Given a query Q, first the set of entity instances that match are looked up in the RDBMS. The query is also sent to every Entity Instance Actor that is active in memory. Each Entity Instance Actor returns a message containing a set comprising of one element (the version of the entity instance that the Actor represents) if the version of the entity matches the query. Alternatively, the message represents the empty set. The final result is then the set of entity instances returned from the DB, removing those that are represented by an Entity Instance Actor where an empty result set was returned, and adding those entity instances which are not returned from the DB but are returned from their Entity Instance Actor.

Algorithm 1 Set Selection Query Algorithm

```
Let $S_1$ = the result of query $Q$ from the RDBMS
Let $S_2$ = the result of query $Q$ executed against the Entity Instance Actors
$result = (S_1 - (\forall s_2 \in S_2$ where $s_2.results = \emptyset) \bigcup ((\forall s_2 \in S_2$ where $s_2.results <> \emptyset))$
```

C. Record Creation Operations

Operations that create a new record are also satisfied by the Thespis API and state-of-the-art CC DBMSs. New records are always created in a causally-consistent context, and therefore the underlying CC+ DBMS is required to manage replication and resolve any potential conflicts. Three queries in the TPC-C benchmark, marked with ¶ in Table II, fall under this segment.

D. Sequence-Management Operations

The query marked with ⊙ is such an operation, which increments a field with every transaction to always ensure a sequential, monotonically increasing, and unique value.

Although this is managed well by an RDBMS, a special operator is needed by a CC DBMS. The following operator is proposed to satisfy such requirements:

$$\eta : \text{uint64} = \overbrace{\{\text{DC id}\}}^{4\text{ bits}}\overbrace{\{\text{physical time}\}}^{48\text{ bits}}\overbrace{\{\text{logical time}\}}^{12\text{ bits}}$$

Operator η is essentially a hybrid logical clock (HLC) [19] that generates monotonically-increasing values at the microsecond granularity tracking of physical time. The value is prefixed by the unique identifier of the DC where the operation is accepted, essentially allowing the CC DBMS to span across

TABLE II
TPC-C TRANSACTIONS AND QUERIES.
† PRIMARY KEY SELECTION. § SET SELECTION. ¶ RECORD CREATION. ⊙ SEQUENCE MANAGEMENT. ⋇ VALUE INCREMENT. ⊎ VALUE ASSIGNMENT. ⊠ RECORD DELETION. ± JOINS, AGGREGATES ETC.

Transaction	Procedure
New Order	1) Select(whouse-id) from Warehouse † 2) Select(dist-id,whouse-id) from District † 3) Update(dist-id,whouse-id) in District ⊙ 4) Select(customer-id,dist-id,whouse-id) from Customer † 5) Insert into Order ¶ 6) Insert into New-Order ¶ 7) For each item (10 items): a) Select(item-id) from Item † b) Select(item-id,whouse-id) from Stock † c) Update(stock-id,whouse-id) in Stock ⋇ d) Insert into Order-Line ¶
Payment	1) Select(whouse-id) from Warehouse † 2) Select(dist-id,whouse-id) from District † 3) a) Case 1: Select(customer-id,dist -id,whouse-id) from Customer † OR b) Case 2: Non-Unique Select(customer-name,dist-id,whouse-id) from Customer § 4) Update(whouse-id) in Warehouse ⋇ 5) Update(dist-id,whouse-id) in District ⋇ 6) Update(customer-id,dist-id,whouse-id) in Customer ⊎ 7) Insert into History ¶
Order Status	1) a) Case 1: Select(customer-id, dist-id,whouse-id) from Customer † b) Case 2: Non-Unique- Select (customer-name, dist-id,whouseid) from Customer § 2) Select (Max (order-id) ,customer-id) from Order ± 3) For each item in the order: a) Select (order-id) from Order-Line †
Delivery	1) For each district within the warehouse (i.e. ten times): a) Select (no-o-id) from New-Order † b) Delete(order-id) from New-Order ⊠ c) Select (customer-id) from Order † d) Update(order-id) from Order ⊎ e) For each item in the order (i.e. ten times): i) Update (delivery-date) from Order-Line ⊎ f) Select (Sum (amount)) from Order-Line ± g) Update(balance) from Customer ⋇ h) Update(delivery-cnt) from Customer ⋇
Stock Level	1) Select (d-next-o-id) from District † 2) Select count(distinct (s-i-id)) from OrderLine,Stock ±

a maximum of 16 DCs, which is deemed sufficient for our use cases. Finally, reserving 12 bits for the logical time part of the HLC, accepting a maximum value of 4096, is also deemed sufficient, based on results that show that this rarely exceeds 100 [19] even in the worst case of clock skew.

This operator satisfies two of the three requirements of the TPC-C workload: it generates a monotonically increasing and unique value at any point in time however, the values are not guaranteed to be sequential. This latter guarantee could be achieved through co-ordination between the DDBMSs nodes, resulting however in the loss of high availability and throughput of the DDBMS. Thus, we see the η operator being useful in the CC DBMS, where application semantics can relax the sequential constraint.

Another approach is a UID generator [48], where part of the acceptable range of values in the sequence is allocated to each DC, that in turn can generate sequential values from its allocation. This is a simpler approach however it assumes that the number of DCs is static and known beforehand, and does not support a cluster that can shrink or grow to the needs of the application.

E. Value-Increment Operations

Trivially, we define value-increment operations as those operations that increment the value of a field in a tuple, such as those marked with ⋇ in Table II. A special operator is also required by a CC DBMS to handle such operations. The following operator is proposed to satisfy such requirements:

$$\gamma(t) : \text{int64} = \sum_{0}^{n-1}[v_0, v_2, v_3, ..., v_{n-1}]$$

Essentially, the operator $\gamma(x)$ yields a *Grow-Only Counter* [49], that stores a value v for every DC participating in the cluster. Executing $\gamma(t)$ at DC x increments the value v_x by t.

Grow-Only Counters are conflict free replicated data types (CRDTs) that support increment operations in a distributed environment without co-ordination [49]. Therefore, they allow a CC DDBMS to execute such operations safely, without introducing co-ordination and losing on high availability and performance.

F. Value-Assignment Operations

Value-assignment operations set an absolute value of a field in a tuple. Such queries, marked with ⊎ in Table II, are equivalent to WRITE operations in a CC DDBMS and do not strictly require any special semantics. However, some OLTP workloads may need to enforce stronger consistency on such operations in order to avoid lost updates, specifically the effects of concurrent operations which are subsequently resolved via the LWW strategy by the CC DDBMS. Such consistency guarantees have been shown to be incompatible with a distributed, highly-available environment [50].

Thespis (with the ThespisDIIP extension [39]) already supports application-specific configuration to enforce invariants that satisfy LAI constraints. We extend this configuration further in order to define fields that require a strong consistency, and therefore co-ordination between nodes. This is in-line with other approaches to multi-level consistency in the literature [51] [52] and we believe that it is an important feature that makes a CC DDBMS applicable to common OLTP workloads where, for some operations, strong consistency is preferred over operation latency or high availability.

G. Record Deletion Operations

Operations that delete records, such as the one marked with ⊠ in the Delivery transaction, are achievable by the same semantics as a WRITE operation of the Thespis API, and therefore inherit the same causal consistency guarantees.

H. Operations involving Joins, Aggregates, Sorting etc.

OLTP queries may comprise other relational algebra operations, such as joins, aggregates and sorting. Such queries in the TPC-C workload are marked with ± in the Stock Level and Order Status transactions.

Join operations can be considered equivalent to a Cartesian product of two relations, followed by a set selection operation [53] and are therefore achievable with the same semantics. Similarly, aggregates and sorting can be implemented in the actor system using a combination of set selection operation semantics and corresponding aggregate or sorting logic. Building on the semantics of operations that have already been defined ensures that CC is guaranteed.

VII. Discussion

The original REST interface of Thespis allows the execution of workloads using a key-value data model. However, we have now shown that the Thespis approach scales to handle also richer data models, such as the relational data model, given that the operators discussed in Section VI are made available in the Thespis API.

With these operators, our analysis shows that it is possible to satisfy the semantics of all the TPC-C transactions in Thespis. Specifically, a client application can interface with Thespis to execute the TPC-C workload in a CC+ distributed environment, thus benefiting from the scalability and high availability properties of a CC+ DBMS.

However, it is important to highlight that Thespis still remains a distributed CC+ DDBMS and therefore foregoes a number of guarantees that a SC RDBMS offers, and that the TPC-C benchmark requires. Most importantly, being a distributed CC+ DBMS, Thespis does not conform with the ACID guarantees that the official TPC-C specification mandates.

The first area of divergence relates to atomicity guarantees. Given that Thespis only supports read-only transactions, via the ThespisTRX extension, multiple WRITE operations are always treated distinctly. Therefore, Thespis does not provide full support for the atomicity criteria that the TPC-C benchmark requires for its transactions.

Furthermore, Thespis guarantees CC, the strongest level of consistency that a DDB can guarantee whilst also supporting high availability. As expected, this level of consistency is not sufficient to guarantee the consistency requirements of TPC-C, which is expected and in-line with orthogonal literature [46].

The isolation requirements of TPC-C are also not adhered to under Thespis, where there is no notion of transactions. A higher level of conformity is guaranteed through the ThespisTRX extension, which protects read-only transactions from phenomena such as phantom and non-repeatable reads. However, the lack of support for WRITE transactions does not prohibit other phenomena such as dirty writes.

Lastly, being a DDB that can accept WRITE operations at multiple DCs, Thespis does not satisfy the durability requirements of TPC-C. Specifically, any WRITE operation (such as record creation, value assignment and record deletion operations) can be overwritten by the LWW conflict resolution operation. Given the possible occurrence of conflicts, and the fact that these cannot be prohibited without losing high availability, the effects of a WRITE operation cannot be deemed sufficiently durable in Thespis to conform with the requirements of the TPC-C benchmark.

VIII. Conclusions

The problem domain is well studied and different approaches have been proposed, including domain specific languages [54], data types that support various consistency levels [55], and instructions expressed in the application's third-generation language [48], [56].

We tackle the problem by proposing a middleware that scales an RDBMS into a CC+ DDBMS, and sits behind an API that is accessible to application developers through an easy interface, abstracting the complexities of CC.

An important contribution of this paper is the analysis of the TPC-C benchmark, a popular OLTP workload, and the proposal of further extensions to the Thespis API that increase the scope of the operations permitted by the CC middleware. We also innovatively show how these extensions allow execution of the OLTP queries in the context of a CC DDBMS whilst retaining an easy-to-use API. We also further analyse the TPC-C workload and highlight areas where Thespis suffers from the intrinsic properties of a CC+ DDB, and therefore does not adhere to the requirements of the TPC-C specification. Nonetheless, given that CC is deemed as a sufficiently strong consistency for enterprise applications, a CC+ DDBMS remains an important tool for such problem domains that are able to forego ACID guarantees for other desirable properties such as scalability and high availability.

Finally, future work consists of the implementation of the semantics proposed in this paper, as well as the execution of suitable benchmarks and a discussion of their respective results. A similar analysis of other transactional workloads, such as the ones mentioned in our review of the literature, is also a future direction of research.

Acknowledgment

This work is partly funded by the ENDEAVOUR Scholarship Scheme (Malta), part-financed by the European Union – European Social Fund ESF under Operational Programme II – Cohesion Policy 2014-2020.

References

[1] E. A. Brewer, "Towards robust distributed systems," in *Proceedings of the nineteenth annual ACM symposium on Principles of distributed computing*, ser. PODC '00, vol. 7, 2000. doi: 10.1145/343477.343502. ISBN 1581131836

[2] S. Gilbert and N. Lynch, "Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services," *ACM SIGACT News*, vol. 33, no. 2, pp. 51–59, 2002. doi: 10.1145/564585.564601

[3] M. P. Herlihy and J. M. Wing, "Linearizability: A correctness condition for concurrent objects," *ACM Transactions on Programming Languages and Systems*, vol. 12, no. 3, pp. 463–492, July 1990. doi: 10.1145/78969.78972

[4] W. Vogels, "Eventually consistent," *Communications of the ACM*, vol. 52, no. 1, pp. 40–44, January 2009. doi: 10.1145/1435417.1435432

[5] L. Lamport, "The part-time parliament," *ACM Transactions on Computer Systems*, vol. 16, no. 2, pp. 133–169, May 1998. doi: 10.1145/279227.279229

[6] M. M. Elbushra and J. Lindström, "Eventual consistent databases: State of the art," *Open Journal of Databases (OJDB)*, vol. 1, no. 1, pp. 26–41, January 2014.

[7] M. Ahamad, G. Neiger, J. E. Burns, P. Kohli, and P. W. Hutto, "Causal memory: Definitions, implementation, and programming," *Distributed Computing*, vol. 9, no. 1, pp. 37–49, 1995. doi: 10.1007/bf01784241

[8] P. Mahajan, L. Alvisi, and M. Dahlin, "Consistency, availability, and convergence," *University of Texas at Austin Tech Report*, vol. 11, 2011.

[9] P. Bailis, A. Ghodsi, J. M. Hellerstein, and I. Stoica, "Bolt-on causal consistency," in *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*. ACM, 2013. doi: 10.1145/2463676.2465279 pp. 761–772.

[10] K. Spirovska, D. Didona, and W. Zwaenepoel, "Optimistic causal consistency for geo-replicated key-value stores," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 3, pp. 527–542, March 2021. doi: 10.1109/tpds.2020.3026778

[11] S. Braun, A. Bieniusa, and F. Elberzhager, "Advanced domain-driven design for consistency in distributed data-intensive systems," in *Proceedings of the 8th Workshop on Principles and Practice of Consistency for Distributed Data*. ACM, April 2021. doi: 10.1145/3447865.3457969 pp. 1–12.

[12] L. Lamport, "Time, clocks, and the ordering of events in a distributed system," *Communications of the ACM*, vol. 21, no. 7, pp. 558–565, 1978. doi: 10.1145/359545.359563

[13] J. C. Corbett, J. Dean, M. Epstein, A. Fikes, C. Frost, J. J. Furman, S. Ghemawat, A. Gubarev, C. Heiser, P. Hochschild *et al.*, "Spanner: Google's globally distributed database," *ACM Transactions on Computer Systems (TOCS)*, vol. 31, no. 3, p. 8, August 2013. doi: 10.1145/2491245

[14] D. B. Terry, M. M. Theimer, K. Petersen, A. J. Demers, M. J. Spreitzer, and C. H. Hauser, "Managing update conflicts in Bayou, a weakly connected replicated storage system," *ACM SIGOPS Operating Systems Review*, vol. 29, no. 5, pp. 172–182, December 1995. doi: 10.1145/224057.224070

[15] A. Lakshman and P. Malik, "Cassandra: a decentralized structured storage system," *ACM SIGOPS Operating Systems Review*, vol. 44, no. 2, pp. 35–40, April 2010. doi: 10.1145/1773912.1773922

[16] W. Lloyd, M. J. Freedman, M. Kaminsky, and D. G. Andersen, "Don't settle for eventual: scalable causal consistency for wide-area storage with cops," in *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*, ser. SOSP '11. Association for Computing Machinery (ACM), 2011. doi: 10.1145/2043556.2043593. ISBN 9781450309776 pp. 401–416.

[17] J. Du, C. Iorgulescu, A. Roy, and W. Zwaenepoel, "Gentlerain: Cheap and scalable causal consistency with physical clocks," in *Proceedings of the ACM Symposium on Cloud Computing*. ACM, November 2014. doi: 10.1145/2670979.2670983 pp. 1–13.

[18] K. Spirovska, D. Didona, and W. Zwaenepoel, "Wren: Nonblocking reads in a partitioned transactional causally consistent data store," in *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, June 2018. doi: 10.1109/dsn.2018.00014 pp. 1–12.

[19] S. S. Kulkarni, M. Demirbas, D. Madappa, B. Avva, and M. Leone, "Logical physical clocks," in *Lecture Notes in Computer Science*, M. K. Aguilera, L. Querzoni, and M. Shapiro, Eds. Springer International Publishing, 2014. doi: 10.1007/978-3-319-14472-6_2 pp. 17–32.

[20] S. Elnaffar, P. Martin, and R. Horman, "Automatically classifying database workloads," in *Proceedings of the eleventh international conference on Information and knowledge management - CIKM '02*. ACM Press, 2002. doi: 10.1145/584792.584898 pp. 622–624.

[21] L. Li, G. Wu, G. Wang, and Y. Yuan, "Accelerating hybrid transactional/analytical processing using consistent dual-snapshot," in *International Conference on Database Systems for Advanced Applications*. Springer International Publishing, 2019. doi: 10.1007/978-3-030-18576-3_4 pp. 52–69.

[22] M. Bach and A. Werner, "Hybrid column/row-oriented DBMS," in *Advances in Intelligent Systems and Computing*. Springer International Publishing, September 2015, pp. 697–707. doi: 10.1007/978-3-319-23437-3_60

[23] N. Poggi, D. Carrera, R. Gavalda, E. Ayguadé, and J. Torres, "A methodology for the evaluation of high response time on e-commerce users and sales," *Information Systems Frontiers*, vol. 16, no. 5, pp. 867–885, October 2014. doi: 10.1007/s10796-012-9387-4

[24] F. Raab, "TPC-C - the standard benchmark for online transaction processing (OLTP)," in *The Benchmark Handbook for Database and Transaction Systems (2nd Edition)*, 1993.

[25] P. Tözün, I. Pandis, C. Kaynak, D. Jevdjic, and A. Ailamaki, "From A to E: analyzing TPC's OLTP benchmarks: the obsolete, the ubiquitous, the unexplored," in *Proceedings of the 16th International Conference on Extending Database Technology - EDBT '13*, 2013. doi: 10.1145/2452376.2452380 pp. 17–28.

[26] S. T. Leutenegger and D. Dias, "A modeling study of the TPC-c benchmark," *ACM SIGMOD Record*, vol. 22, no. 2, pp. 22–31, June 1993. doi: 10.1145/170036.170042

[27] T. P. P. C. TPC, "Tpc benchmark e," 2010.

[28] S. Chen, A. Ailamaki, M. Athanassoulis, P. B. Gibbons, R. Johnson, I. Pandis, and R. Stoica, "TPC-E vs. TPC-C: Characterizing the new TPC-E benchmark via an I/O comparison study," *ACM SIGMOD Record*, vol. 39, no. 3, pp. 5–10, February 2011. doi: 10.1145/1942776.1942778

[29] M. J. Cahill, U. Röhm, and A. D. Fekete, "Serializable isolation for snapshot databases," *ACM Transactions on Database Systems*, vol. 34, no. 4, pp. 1–42, December 2009. doi: 10.1145/1620585.1620587

[30] C. Camilleri, J. G. Vella, and V. Nezval, "Thespis: Actor-Based Causal Consistency," in *Database and Expert Systems Applications (DEXA), 2017. 28th International Workshop on Big Data Management in Cloud Systems*. IEEE, August 2017. doi: 10.1109/dexa.2017.25 pp. 42–46.

[31] C. Hewitt, P. Bishop, and R. Steiger, "A universal modular actor formalism for artificial intelligence," in *Proceedings of the 3rd international joint conference on Artificial intelligence*. Morgan Kaufmann Publishers Inc., 1973, pp. 235–245.

[32] G. Agha, *Actors: A Model of Concurrent Computation in Distributed Systems*. The MIT Press, 1986. doi: 10.7551/mitpress/1086.001.0001

[33] G. Young, "CQRS documents by Greg Young," 2010. [Online]. Available: https://github.com/keyvanakbary/cqrs-documents

[34] B. Meyer, *Eiffel: The Language*. Prentice-Hall, Inc., December 1992. ISBN 0-13-247925-7. doi: 10.1016/0950-5849(92)90131-8

[35] M. Fowler, "Event sourcing," December 2005. [Online]. Available: https://martinfowler.com/eaaDev/EventSourcing.html

[36] D. Abadi, “Consistency tradeoffs in modern distributed database system design: CAP is only part of the story,” *Computer*, vol. 45, no. 2, pp. 37–42, February 2012. doi: 10.1109/mc.2012.33

[37] C. Camilleri, J. G. Vella, and V. Nezval, “ThespisTRX: Causally-consistent read transactions,” *International Journal of Information Technology and Web Engineering (IJITWE)*, vol. 15, no. 1, pp. 1–16, January 2020. doi: 10.4018/ijitwe.2020010101

[38] V. Balegas, S. Duarte, C. Ferreira, R. Rodrigues, and N. Preguiça, “IPA: Invariant-preserving applications for weakly-consistent replicated databases,” *Proceedings of the VLDB Endowment*, vol. 12, no. 4, pp. 404–418, December 2018. doi: 10.14778/3297753.3297760

[39] C. Camilleri, J. G. Vella, and V. Nezval, “ThespisDIIP: Distributed integrity invariant preservation,” in *International Conference on Database and Expert Systems Applications*. Springer International Publishing, 2018. doi: 10.1007/978-3-319-99133-7_2 pp. 21–37.

[40] D. Barbará-Millá and H. Garcia-Molina, “The demarcation protocol: A technique for maintaining constraints in distributed database systems,” *The VLDB Journal - The International Journal on Very Large Data Bases*, vol. 3, no. 3, pp. 325–353, July 1994. doi: 10.1007/bf01232643

[41] N. Krishnakumar and A. J. Bernstein, “High throughput escrow algorithms for replicated databases,” ser. VLDB ’92. Morgan Kaufmann Publishers Inc., 1992. ISBN 1558601511 pp. 175–186.

[42] P. Bailis, A. Fekete, M. J. Franklin, A. Ghodsi, J. M. Hellerstein, and I. Stoica, “Feral concurrency control: An empirical investigation of modern application integrity,” in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM, May 2015. doi: 10.1145/2723372.2737784 pp. 1327–1342.

[43] P. Bailis, A. Fekete, M. J. Franklin, and A. Ghodsi, “Coordination avoidance in database systems,” *Proceedings of the VLDB Endowment*, vol. 8, no. 3, pp. 185–196, 2014. doi: 10.14778/2735508.2735509

[44] N. Soparkar and A. Silberschatz, “Data-valued partitioning and virtual messages,” in *Proceedings of the ninth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems - PODS '90*. ACM Press, 1990. doi: 10.1145/298514.298587 pp. 357–367.

[45] B. F. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Sears, “Benchmarking cloud serving systems with ycsb,” in *Proceedings of the 1st ACM symposium on Cloud computing - SoCC '10*. ACM Press, 2010. doi: 10.1145/1807128.1807152 pp. 143–154.

[46] K. Rahmani, K. Nagar, B. Delaware, and S. Jagannathan, “CLOTHO: directed test generation for weakly consistent database systems,” *Proceedings of the ACM on Programming Languages*, vol. 3, no. OOPSLA, pp. 1–28, October 2019. doi: 10.1145/3360543

[47] A. Chikhaoui, K. Boukhalfa, and J. Boukhobza, “A cost model for hybrid storage systems in a cloud federations,” in *Proceedings of the 2018 Federated Conference on Computer Science and Information Systems (FedCSIS)*, ser. Annals of Computer Science and Information Systems, M. Ganzha, L. Maciaszek, and M. Paprzycki, Eds., vol. 15. IEEE, September 2018. doi: 10.15439/2018F237 pp. 1025–1034.

[48] V. Balegas, S. Duarte, C. Ferreira, R. Rodrigues, N. Preguiça, M. Najafzadeh, and M. Shapiro, “Putting consistency back into eventual consistency,” in *Proceedings of the Tenth European Conference on Computer Systems*, April 2015. doi: 10.1145/2741948.2741972 pp. 1–16.

[49] M. Shapiro, N. Preguiça, C. Baquero, and M. Zawirski, “A comprehensive study of convergent and commutative replicated data types,” Ph.D. dissertation, Inria–Centre Paris-Rocquencourt; INRIA, 2011. [Online]. Available: https://hal.inria.fr/inria-00555588

[50] P. Bailis, A. Davidson, A. Fekete, A. Ghodsi, J. M. Hellerstein, and I. Stoica, “Highly available transactions: Virtues and limitations,” *Proceedings of the VLDB Endowment*, vol. 7, no. 3, pp. 181–192, November 2013. doi: 10.14778/2732232.2732237

[51] C. Li, D. Porto, A. Clement, J. Gehrke, N. Preguiça, and R. Rodrigues, “Making geo-replicated systems fast as possible, consistent when necessary,” in *10th USENIX Symposium on Operating Systems Design and Implementation (OSDI’12)*, 2012. ISBN 9781931971966 pp. 265–278.

[52] A. Bouajjani, C. Enea, M. Mukund, G. Shenoy, and S. Suresh, “Formalizing and checking multilevel consistency,” in *International Conference on Verification, Model Checking, and Abstract Interpretation*. Springer International Publishing, 2020. doi: 10.1007/978-3-030-39322-9_18 pp. 379–400.

[53] R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, 7th ed. Pearson, June 2015. ISBN 0133970779

[54] M. Milano and A. C. Myers, “Mixt: A language for mixing consistency in geodistributed transactions,” in *Proceedings of the 39th ACM SIGPLAN Conference on Programming Language Design and Implementation*, vol. 53, no. 4. ACM, June 2018. doi: 10.1145/3192366.3192375 pp. 226–241.

[55] B. Holt, J. Bornholt, I. Zhang, D. Ports, M. Oskin, and L. Ceze, “Disciplined inconsistency with consistency types,” in *Proceedings of the Seventh ACM Symposium on Cloud Computing*, October 2016. doi: 10.1145/2987550.2987559 pp. 279–293.

[56] M. Köhler, N. Eskandani, P. Weisenburger, A. Margara, and G. Salvaneschi, “Rethinking safe consistency in distributed object-oriented programming,” *Proceedings of the ACM on Programming Languages*, vol. 4, no. OOPSLA, pp. 1–30, November 2020. doi: 10.1145/3428256

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 271–279

DOI: 10.15439/2021F86
ISSN 2300-5963 ACSIS, Vol. 25

Enabling Autonomous Medical Image Data Annotation: A human-in-the-loop Reinforcement Learning Approach

Leonardo C. da Cruz, César A. Sierra-Franco, Greis Francy M. Silva-Calpa, Alberto Barbosa Raposo
Department of Informatics
Tecgraf Institute
Pontifical Catholic University of Rio de Janeiro (PUC-Rio)
Gávea, 22451-900, Rio de Janeiro, Brazil
Email: {lccruz,casfranco,greis,abraposo}@tecgraf.puc-rio.br

Abstract—**Deep learning techniques have shown significant contributions to several fields, including medical image analysis. For supervised learning tasks, the performance of these techniques depends on a large amount of training data as well as labeled data. However, labeling is an expensive and time-consuming process. With this limitation, we introduce a new approach based on Deep Reinforcement Learning (DRL) to cost-effective annotation in a set of medical data. Our approach consists of a virtual agent to automatically label training data, and a human-in-the-loop to assist in the training of the agent. We implemented the Deep Q-Network algorithm to create the virtual agent and adopted the method mentioned above, which employs human advice to the virtual agent. Our approach was evaluated on a set of medical X-ray data in different use cases, where the agent was required to create new annotations in the form of bounding boxes from unlabeled data. Results show that an agent training with advice positively impacts obtaining new annotations from a data set with scarce labels. This result opens up new possibilities for advancing the study and implementing autonomous approaches with human advice to create a cost-effective annotation in data sets for computer-aided medical image analysis.**

I. Introduction

ARTIFICIAL Intelligence (AI) techniques, mainly those based on supervised learning, require a large amount of annotated data for training a model. In intelligent systems for the health field, the use of these techniques has contributed to the processing and analysis of medical images [1] [2]; however, the absence of labeled data has been a limitation for the implementation of those solutions.

Annotated data is necessary to enable the network to learn the relationship between a desired input and output during a machine learning model training. With sufficient data and annotation, the accuracy of a model often corresponds to or exceeds the level of expert physicians in classifying and detecting diseases [3]. However, obtaining new annotations is an expensive and time-consuming task. That labeling process is often performed manually by human experts. To reduce efforts at annotations, researchers have explored approaches of cost-effective data annotation [4]. An example of this approach are Active Learning algorithms. These algorithms aim to reduce the cost of labeling, selecting only the images to be labeled by the human, which are informative to improve the accuracy of a model [5].

However, the active learning algorithm still needs the human to make annotations of data. This aspect motivates the development of this study, contributing to creating an approach to automatically label data.

We present an approach that aims to contribute to scarce annotations based on a cost-effective data annotation approach. In particular, we focus on creating new annotations automatically on medical examinations, reducing the time and cost of the annotations. To meet the proposal, we use two objectives: 1) use of the Reinforcement Learning (RL) algorithms [6]: for creating an autonomous virtual agent. 2) insertion of the human in the training process: to teach the autonomous agent to perform its task correctly even with scarce annotations.

Reinforcement Learning (RL) is a machine learning paradigm that consists of how a virtual agent (we will adopt the term RL agent) finds a solution to a given problem, exploring interactions in the environment. Mnih et al [7] proposed Deep Reinforcement Learning (DRL) that combines RL and Convolutional Neural Network (CNN). This model is a CNN trained with a variant of the RL algorithm called Q-Learning. This method aims to enable the connection between an RL algorithm and deep neural network algorithms, operating on images with raw pixels.

In recent years, DRL models have achieved advances that surpass human performance in games such as Atari [8], has also demonstrated promise in enabling physical robots to learn complex skills in the real world [9] and in real world deployment of autonomous driving [10]. Traditionally, DRL has employed one type of algorithm that is Deep Q-Network (DQN) [7] [11].

Some authors, such as Son and Gong [12], and Liu, et al.[13] have proposed resolving the problem of scarce annotations using DQN algorithm to automate the selection process of unlabeled data. With this, an RL agent learns a data selection criterion; however, they still require the participation of a human for the labeling process. Our study shows an RL agent for automatic labeling, where we include the human in the training loop of the RL algorithm. This inclusion is due to the human's ability to teach tasks, evaluate performance, intervene at certain times to avoid unwanted actions, and increase the RL agent's learning efficiency.

In summary, this study presents the following contributions:

1) A new approach to reduce efforts to acquire new annotated data.
2) Integration of the human to speed up the learning of RL agent contributing to efficiency in creating new annotations.

The rest of this paper is organized as follows. In Section 2, we present the related work. In Section 3, we detail the proposed approach, which includes a description of reinforcement learning, the steps for understanding the Deep Q-Network (DQN), followed by the implementation of algorithms and the methods of advice. The evaluation and experimental results are described in Section 4 and 5. Finally, Section 6 shows some concluding remarks and future research perspectives.

II. Related Work

In this section, we describe some related studies that use RL algorithms to solve the scarce annotations problem through a cost-effective annotation approach. Also, we present some studies that integrate the human in the training process of these algorithms.

A. Cost-Effective Annotation

Currently, a considerable amount of medical data is available, however, the use of those data without sufficient labels or annotations is a problem when applications use supervised learning methods. Cost-effective annotation approaches are an important strategy to obtain additional annotations in a quick way, and avoiding high costs.

Saripalli et al.[14] present an approach to contribute with the labeling process where data from health monitoring devices need to be interpreted. The authors used RL algorithms to create an RL agent capable of annotating alarm data based on the annotations made by a specialist. As a result, the approach presented by the authors has created mock medical domain experts with high sensitivity, while still catching a notable number of false alarms.

Wang et al. [15] present the Deep Reinforcement Learning Active (DRLA), a new method for medical image classification. This method uses the DQN algorithm applied with the actor-critic paradigm to create an agent capable of learning a more informative image selection policy to be annotated by a human. The method presented a practical approach to relieve human efforts in making annotations.

Zimo et al.[13] proposed another approach using active learning called Deep Reinforcement Active Learning (DRAL). The objective of the study is to minimize human efforts to obtain annotation. Applied in the case of re-identification, the RL agent learns to select the best pair of images for the human annotator, which will give binary feedback to label the image as right or wrong. With each input from the human, a reward is given to the agent.

Sun and Gong [12] also present a new framework that uses active learning to annotate images. They proposed a structure that uses DRL as a data selection strategy. Instead of choosing which image to annotate using heuristic algorithms, the RL algorithm learns a selection policy. The authors evaluated the method with other studies of state of the art, which obtained superior results in a set of popular data.

Other studies address the making of automatic annotations, as a method based only on active learning [16] where the proposed method improved the classification performance compared to the baselines, in a tangent vector of the contour of the image [17]. In the present paper, the proposed method can greatly reduce the annotation time while obtaining the same or a higher annotation quality and through interaction [17].

B. Human-in-the-loop Reinforcement Learning

The inclusion of human-in-the-loop for the training of an RL agent is influenced by the human's ability to teach tasks, evaluate performance, and intervene at certain times to avoid destructive actions. This inclusion can increase the speed of the RL agent, making it confident to make quick and accurate decisions, as highlighted by Liang et al [18].

Torrey and Taylor [19] proposed an advice approach called action advice, where a human teacher suggests the student agent's actions to achieve its goal. With a fixed number of times that the human can advise, the authors present algorithms for different moments of counseling, which they call early advising, importance advising, mistake correcting, and predictive advising.

Lin et al. [20] present a method to analyze the performance of action advice in a DRL algorithm. They use human feedback to improve the performance of the RL agent through advice. This method uses an arbiter, which decides when to use actions generated by the policy of the DRL algorithm or actions advised by the human subject.

Krening [21] presents a study investigating whether human insertion as a teacher brings benefits to the student agent. As a contribution of that study, two algorithms for human interaction that promote positive experiences are presented, the Newtonian Action Advice and Object-Focused advice.

Another alternative presented in the literature to human-in-the-loop is modeling the reward that the RL agent will receive after performing actions. Denominated reward shaping, this method uses human feedback as a reward function. We find studies by Knox [22] and Arakawa [23], which show methods to train RL agents with humans as a reward function.

In the literature, there are other proposals to integrate the human in the training process of a DRL algorithm, such as by demonstration [24], imitation [25], and heuristic methods to select a state where the human subject should send actions to the RL system, as shown in the study by [26]. However, these methods need further investigation for agent training. Our approach aims to create an RL agent capable of creating new annotations from few human interactions, thus reducing the cost of generating new annotations.

Table I shows a comparison between our study and studies in the literature.

III. Proposed approach

We integrate the human in the training loop to contribute to the learning process of the RL agent. With this, the agent can generate a more significant number of annotations from a few annotated data samples. Hence, supervised convolutional neural networks could take advantage of an increased machine learning ready dataset for training purposes.

As shown in Figure 1, a problem that extends from dataset limitations are scarce annotations. Some strategies are adopted in the literature with some solutions to this problem, such as Data Augmentation, Leveraging External Labeled Datasets, Cost-Effective Annotation, Leveraging Unlabeled and Regularized Training. Based on the strategies of cost-effective annotation, we present an approach to reduce efforts to acquire new annotated data, creating an RL agent that does this task automatically.

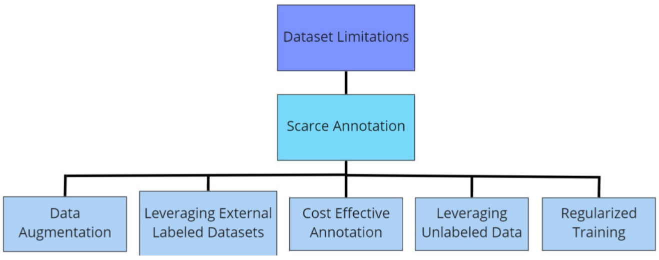

Dataset Limitations
Scarce Annotation
Data Augmentation
Leveraging External Labeled Datasets
Cost Effective Annotation
Leveraging Unlabeled Data
Regularized Training

Fig. 1. Organization of strategies that can be used based on the problem of scarce annotations (image adapted from [27]).

A. Background

Q-Learning is a classic algorithm for reinforcement learning implementation. This algorithm is an off-policy Temporal Difference that focuses on state-action value. The action value in each state is obtained using a table that is updated in each interaction with the environment, denoted Q-values, as shown in the equation 1.

$$Q(s,a) = Q(s,a) + \alpha[r + \gamma.maxQ(s',a') - Q(s,a)] \quad (1)$$

where s is the current state and a is an action taken in this state. When each action a is taken, a new s' state is selected, and a reward issued for that pair of *(s, a)*. For the new selected state, a new action *a'* is taken, chosen randomly using a predefined probability (a method called Epsilon-Greedy Policy). α is the learning rate, *r* is the reward for an action taken in a given state and γ and the factor of discount.

With the success of this classic algorithm, Mnih et al [7] proposed combining Q-Learnig and Convolutions Neural Network (CNN) and presented a algorithm called Deep Q-Network.

B. Understanding Deep Q-Network

We used the DQN algorithm for the agent learning process. It uses a neural network with convolutional neural networks (CNN) to approximate the Q value of all possible actions in each state. Two techniques are the pivot for the success of this algorithm: experience replay and target network.

1) Experience replay: It serves to store the experiences acquired by the RL agent at each step. A memory buffer was used to store a predetermined amount of past experiences (batch size). At each step t, a transition is saved in this memory buffer and then used to train the neural network via stochastic gradient descent.

A transition is a tuple formed by the Markov Decision Process (MDP), where it is composed of an MPD tuple (S, A, R, S '), being:

- *S (State)*: The current state.
- *A (action)*: Action performed in the current state.
- *R (reward)*: Reward for an action taken in a given state.
- *S' (Next State)*: Next state.

Figure 2 illustrates the storage of transitions in a memory buffer;

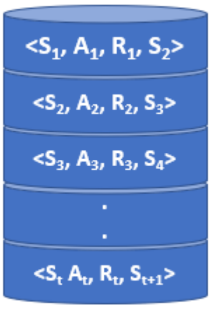


Fig. 2. Experience replay storage illustration in DQN algorithms.

2) Target Network: The Loss equation calculates the difference between the target and the prediction value, as shown in Equation 2. DQN uses a second neural network called target network to optimize the loss equation and calculate the target value.

$$Loss = (r + \gamma max_{a'} Q(s', a'; \Theta) - Q(s, a; \Theta))^2 \quad (2)$$

The Target network is a clone of the policy network and its used to calculate the target value. Initially, their weights are frozen with the weights of the original policy net and are updated with the new weights of the policy net for a certain period. The loss function given by,

$$Loss = (r + \gamma max_{a'} Q(s', a'; \Theta') - Q(s, a; \Theta))^2 \quad (3)$$

where:

- r = reward
- γ = discount factor
- Θ' = Is updated weights once every target steps.
- Θ = Learns the correct weights by using gradient descent

TABLE I
COMPARISON TABLE BETWEEN OUR STUDY AND RELATED WORKS

Reference	**algorithm**	**Medical Aplication**	**HRL Method**
V. R. Saripall et al.[14]	DQN, A2C	Annotate medical signal data	N/A
J. Wang et al.[15]	DQN, AL	Image classification	N/A
Z.Liu, et al.[13]	RL, CNN, AL	N/A	Policy Shaping
Sun and Gong [12]	DQN, AL	N/A	N/A
Torrey e Taylor [19]	SARSA, Q-LEARNING	N/A	Advice
Lin, et al.[20]	DQN	N/A	advice
Krening [21]	BQL	N/A	advice
Knox [22]	Supervised Learning and RL	N/A	Reward shaping
Our study	**DQN**	**Automatic annotation in x-ray images**	**advice**

C. Implementation of the Deep Q-Network algorithm

Based on study of Caicedo et al. [28], [29], we started by implementing the DQN algorithm to locate objects in two-dimensional (2D) images.

At each step, the RL agent observes the current state (region of an image) and estimates the potential rewards based on the cost of taking different actions. After this calculation, it selects the action that will lead it to receive the maximum reward and moves on to the next state. This process is repeated until it reaches the terminal state. This cycle within the RL is called an episode. The following is a mapping of the MDP to the context of our work.

1) States: A medical image represents a state within our context of locating a desired region. The RL agent's area visualization is of the image size and will serve as input data for the network. At each step of the algorithm, the agent analyzes pixels of the image within its viewing area and thus calculates the best action to be taken. With each action performed by the RL agent, its viewing area will be adjusted until the object of interest is located. The next state is the current image, and the agent's viewing area is adjusted by the last action taken. The terminal state is when the agent stops performing actions because it has already completed its search. In this case, is create a new bounding box if was found an object.

2) Actions: We adopted a set of nine actions that agent RL can perform in the current state, were applied eight of which to the deformation of the agent's viewing area and one to indicate the terminal state, as shown in Figure 3. As the agent takes his actions, the agent's bounding box is deformed until it fits in the space of the object of interest.

Figure 5 illustrates the actions that the RL agent takes to detect a region of interest.

3) Rewards: The reward function used for this work is the same as presented by Caicedo et al. [28].

Equation 4 is calculated to assign rewards to the RL agent for each action taken. This equation is formed by the current visualization area of the agent RL b, together with the ground truth of the target object to be located g, and b' is the visualization area in the next step. In general, this function will attribute a positive reward to the agent if the action taken improves the IoU between the current and the next state, otherwise, the reward will be negative, as Equation 5 represents.

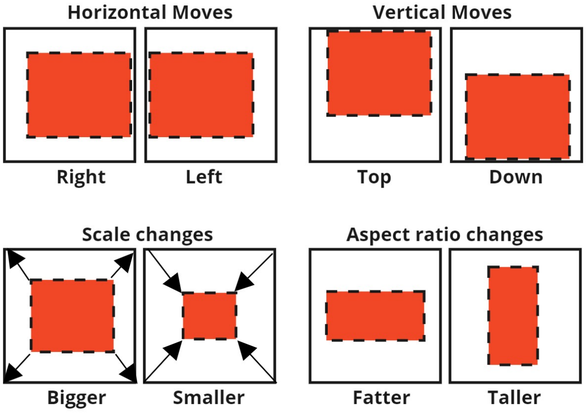


Fig. 3. Illustration of the actions that the RL agent perform in the States.

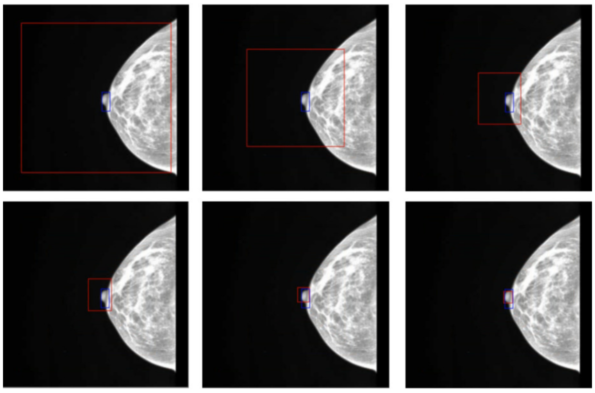


Fig. 4. Image illustrating agent RL creating an annotation in the form of the bounding box of the papilla in a mammography exam.

$$RewSign_a(s,s') = sign(IoU(b',g) - IoU(b,g)) \quad (4)$$

$$\begin{cases} +1, & \text{if } RewSign_a(s,s') > 0 \\ -1, & \text{Otherwise} \end{cases} \quad (5)$$

TABLE II
LEARNING HYPERPARAMETERS

Parameter	Value
Target network update	10000
Replay memory size	50000
Number of episodes	5
Discount factor	0.99
Learning steps	700
Leaning rate	0.00025
Epsilon start	1.0
Epsilon end	0.2
Batch size	32
Optimizer	RMSProp

Equation 6 rewards the agent when it reaches the terminal state according to the final result. In this case, we check if the IoU is greater than or equal to the threshold t (we adopt $t = 0.3$ and 0.5, depending on the use case). With that, the agent receives a positive or negative reward.

$$\begin{cases} +3, & \text{if } IoU \geq t \\ -3, & \text{Otherwise} \end{cases} \tag{6}$$

4) Hyperparameters: Table II sumarize the hyperparameters used for training the RL agent.

D. DQN architecture

DQN architecture uses a sequence of layers of a convolutional network to extract features of the image. The input to the network will be the raw frame of an image. It's common to downsample the pixel and convert the RGB values to grayscale values to reduce computation and consume less memory. Fully connected layers are used with an activation function to estimate Q values directly from the image. The last layer defines the number of units of the output layer according to the possible actions in the environment.

The following diagram shows the DQN architecture used:

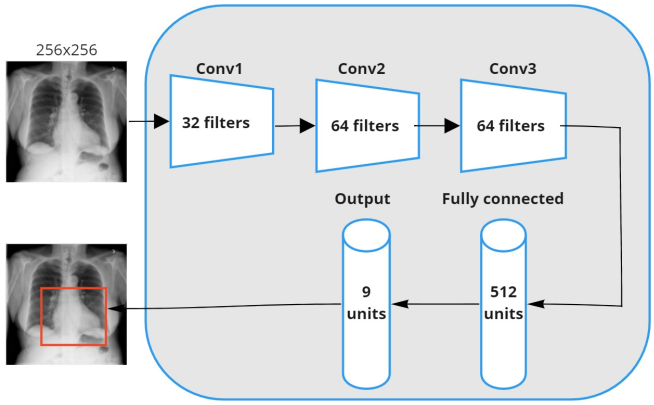


Fig. 5. Architecture used for the DQN algorithm. The input is an image with 256 x 256 pixels and processed by convolutional layers. The output layer predicts the value for the nine possible actions to be taken by the agent.

Algorithm 1 Algorithm for advice

```
Require: Medical image
Ensure: bouding box annotation
  for each episode do
    budget = 5;
    for each state do
      Calculates uncertainty;
      if uncertainty >= 1.2 then
        if budget > 0 then
          Aagent receives human advice
          budget = budget - 1
        else
          The agent takes action generated by your policy.
        end if
      end if
    end for
  end for
```

E. Implementation of an advice method

As an initial experiment, we adopted the method called early advising proposed by Torrey and Taylor [19]. The idea of this method is that the initial states are essential to the advise process, as they have a grater impact in the agent learning process. We adopted a limit of 5 pieces of advice that the human teacher can apply per episode. Algorithm 1 represents the pseudocode of the implemented method.

The human goes on to advise the RL agent when it has uncertainty about what action to take. For this experiment, a threshold of 1.2 was set, since, after a visual observation, we detected that the RL agent tends to take suitable actions below this value. As an experimental phase, the user informs the suggested action through the keyboard, inserting numbers that correspond to the agent's actions.

- Move right = 0
- Move down = 1
- Scale Bigger = 2
- Aspect ratio Fatter = 3
- Move left = 4
- Move up = 5
- Scale Smaller = 6
- Aspect ratio Taller = 7
- Trigger = 8

IV. Evaluation

As suggested by Poole and Mackworth [30], a way to measure an agent's performance is by analyzing the cumulative reward per episode. As the RL agent learns to perform the actions correctly, it receives increased rewards.

We also evaluate quantitatively the agent performance through the number of annotations that it was able to make with and without human help. In addition, we adopt metrics such as Intersection Over Union (IoU) and Average Precision (AP).

The IoU is an evaluation metric used to measure the accuracy of an object detector on a specific data set. It is a measure of the overlap between two areas, that of the bounding box generated by the algorithm and the ground-truth bounding box [31].

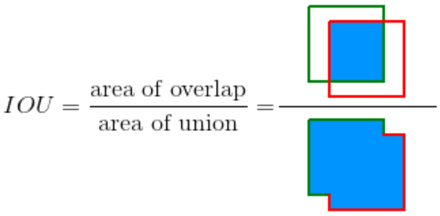

$IOU = \frac{\text{area of overlap}}{\text{area of union}} =$

Fig. 6. The image illustrates a ground truth bounding box (in green) and a bounding box generated by a model (in red). Source [31]

Through a threshold t, the IOU allows one to classify whether the detection of the object is correct ($IOU >= t$) or incorrect ($IOU < t$). This implies that if the IOU is greater than or equal to the threshold, the *bouding box* created is within the expected (TP - True Positive). Otherwise, the created *bouding box* is lower than expected (FP - False positive).

Average Precision (AP) is the metric used to measure the model's ability to identify only the object of interest. The result ranges from 0 to 1. The closer to 1, the more accurate the model will be in creating new annotations.

After implementing our algorithm and the definitions of the evaluation metrics, we applied our approach in two different use cases.

A. Use case 1: Chest examination database

We started by analyzing the agent's performance with and without advice in a database with chest X-ray medical exams for cardiomegaly detection [32]. For this purpose, we use the chest X-ray database from NIH [33]. Cardiomegaly refers to an enlarged heart condition. It is one of the most common inherited diseases of cardiovascular diseases with a prevalence of at least 1 in 500 in the general population [34] [35].

Chest X-ray examinations are frequent and economical. However, the clinical diagnosis of a chest X-ray can be challenging and sometimes more complex than the diagnosis by chest computed tomography. The lack of large, publicly available data sets with meaningful annotations is challenging, delaying the detection and diagnosis of chest X-ray examinations.

B. Use case 2: Mammography exam database

A second use case, which we tested our approach, was in cases of mammography images. Breast cancer can be considered one of the most common global health problems and is considered the second leading cause of cancer mortality in women [36] [37].

Breast images are acquired through an x-ray examination. Two projections are made during the examination procedure: the Cranial Caudal (CC) and Medio Lateral Oblico (MLO) planes. In the CC view, the breast is seen from top-down, while in the MLO, the view is from the lateral region.

TABLE III
TRAINING DATA OF CARDIOMEGALY

Experiments	Advice	# Images	Pre-trained	# Annotations
exp1	No	31	No	11
exp2	Yes	31	No	17
exp3	No	31	Yes	17
exp4	**Yes**	**31**	**Yes**	**19**

TABLE IV
TEST DATA OF CARDIOMEGALY

Experiments	Advice	# Images	Pre-trained	# Annotations	AP
exp1	No	64	No	25	0.3
exp2	**Yes**	**64**	**No**	**38**	**0.5**
exp3	No	64	Yes	37	0.5
exp4	Yes	64	Yes	32	0.4

The nipple is a structure of interest to be observed in mammography exams. This structure helps the mammography technician verify the quality of the positioning of an exam, which can minimize the need for patients to return to repeat the exam caused by poor positioning [38]. However, detecting this structure is not trivial since, in addition to being a small structure, it does not always appear clearly in the images.

V. Experimental Results

A. Use case 1: Chest examination database

We conducted four training experiments with the RL agent to analyze its performance in taking notes automatically. The description of the data used for training is highlighted in Table III.

Table IV presents the results obtained on a set of unlabeled tests.

Figure 7 shows the evolution of the learning of the RL agent when creating annotations the structure of cardiomegaly. Throughout the episodes (indicated by the horizontal axis), is shown the accumulation of expenses (vertical axis) that the RL agent obtained. Negative rewards signify that the RL agent had a hard time learning how to take notes.

As shown in Figure 7, and Table IV, with the insertion of the human in the training loop, the agent was able to obtain better results compared to training without advice, where his learning oscillated more.

Figure 8 illustrates the result obtained by the RL agent when creating a new annotation in the form of a bounding box. The model used was the one that presented the best result, that is, the advice with a **AP = 0.5**.

B. Use case 2: Mammography exam database

Likewise, for this use case, we have carried out four training experiments. The description of the data used can be seen in Table V. The RL agent was trained to automatically create new notes of the nipple from exams projected on the CC plane (Cranio Caudal).

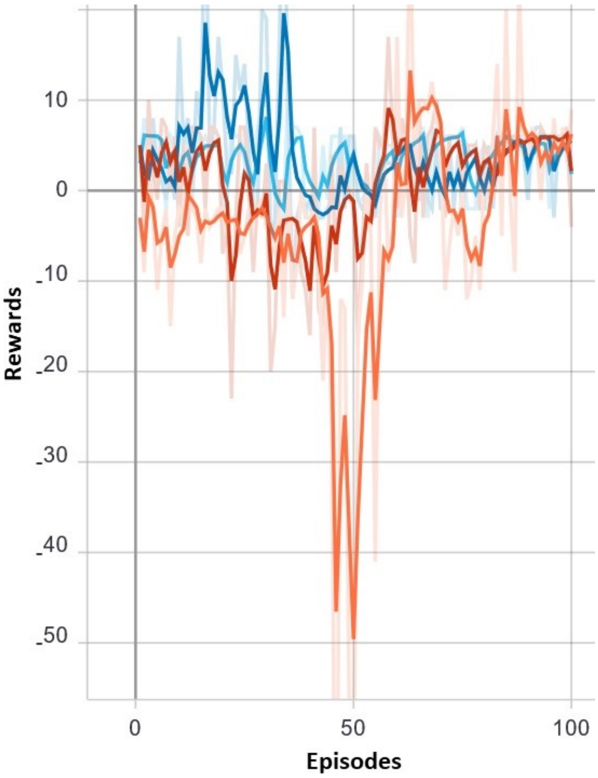


Fig. 7. Result of training of the RL agent to detect the structure of cardiomegaly. Different colors are highlighting the comparison between the experiments.

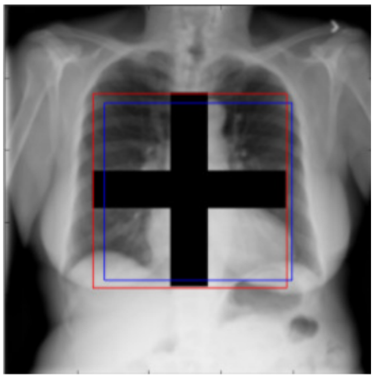


Fig. 8. Cardiomegaly image with being detected. In blue the ground truth, and red the bounding box generated by the agent.

TABLE V
TRAINING DATA OF NIPPLE

Experiments	Advice	# Images	Pre-trained	# Annotations
exp1	No	31	No	0
exp2	Yes	31	No	5
exp3	No	31	Yes	1
exp4	**Yes**	**31**	**Yes**	**15**

TABLE VI
TEST DATA OF NIPPLE

Experiments	Advice	# Images	Pre-trained	# Annotations	AP
exp1	No	192	No	0	0.003
exp2	Yes	192	No	34	0.16
exp3	No	192	Yes	6	0.03
exp4	**Yes**	**192**	**Yes**	**60**	**0.3**

We performed the RL agent testing experiments from a database without annotations. Table VI presents the results obtained.

Figure 9 shows the evolution of the learning of the RL agent when creating annotations of a region of interest to the breast. Throughout the episodes (indicated by the horizontal axis), it is shown the accumulation of expenses (vertical axis) that the RL agent obtained. Negative rewards signify that the RL agent had a hard time learning how to take notes. As the graph shows, the experiment that presented the best rewards, i.e., the agent obtained positive rewards, was through apprenticeship learning.

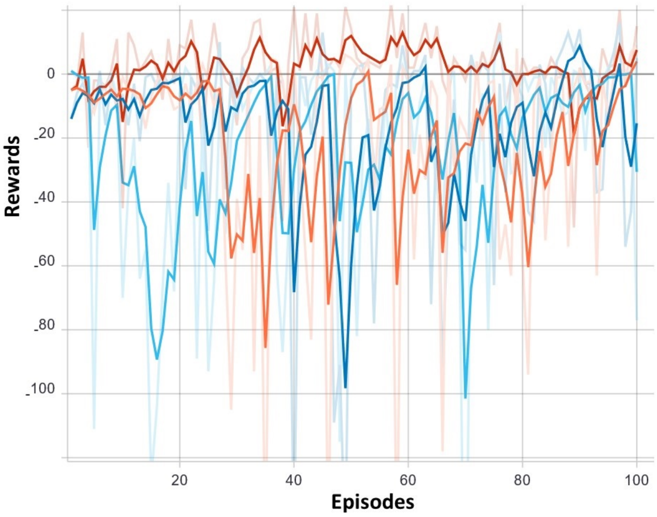


Fig. 9. Result of training of the RL agent to detect the structure of the papilla. Different colors are highlighting the comparison between the experiments.

As shown in Figure 9 and Table VI, training the RL agent with advice impacts positively in creating new annotations automatically. On the other hand, the RL agent, without counseling, proved to be less effective, having difficulties in learning the task.

Figure 10 illustrates the result obtained by the RL agent when creating a new annotation in the form of the papilla's bounding box. The model used was the one that presented the best result, that is, the advice with a **AP = 0.3**.

VI. Conclusions

This paper presents a new approach for a cost-effective annotation in a set of medical data, where annotations are performed in an automated manner by a virtual agent through

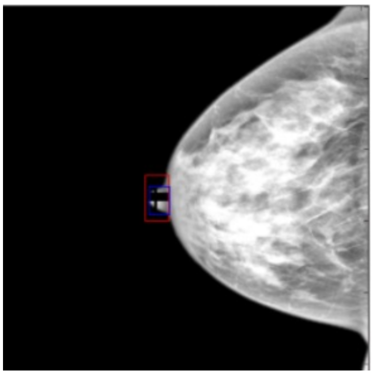


Fig. 10. Nipple image with being detected. In blue the ground truth, and red the bounding box generated by the agent.

human advice. We evaluated our approach in medical datasets for chest and mammography X-ray. The results showed that the human advice allowed the RL agent to perform learning even with a small sample of annotated data. The results also showed how early human assistance increased both precision and convergence speed to the annotation learning process.

For future work, we plan to perform experiments adjusting a more significant number of hyperparameters, analyze the amount of advice given by the human, and advise at different times during the agent training process. In addition, we intend to implement an active learning approach to increase the autonomous agent accuracy, increasing its capacity to create new annotations suitable for supervised machine learning algorithms.

REFERENCES

[1] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. Van Der Laak, B. Van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical image analysis*, vol. 42, pp. 60–88, 2017. [Online]. Available: https://doi.org/10.1016/j.media.2017.07.005

[2] A. Esteva, K. Chou, S. Yeung, N. Naik, A. Madani, A. Mottaghi, Y. Liu, E. Topol, J. Dean, and R. Socher, "Deep learning-enabled medical computer vision," *NPJ digital medicine*, vol. 4, no. 1, pp. 1–9, 2021. [Online]. Available: https://doi.org/10.1038/s41746-020-00376-2

[3] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *nature*, vol. 542, no. 7639, pp. 115–118, 2017. [Online]. Available: https://doi.org/10.1038/nature21056

[4] J. Yang, J. Fan, Z. Wei, G. Li, T. Liu, and X. Du, "Cost-effective data annotation using game-based crowdsourcing," *Proceedings of the VLDB Endowment*, vol. 12, no. 1, pp. 57–70, 2018. [Online]. Available: https://doi.org/10.14778/3275536.3275541

[5] S. Budd, E. C. Robinson, and B. Kainz, "A survey on active learning and human-in-the-loop deep learning for medical image analysis," *Medical Image Analysis*, p. 102062, 2021. [Online]. Available: https://doi.org/10.1016/j.media.2021.102062

[6] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[7] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[8] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015. [Online]. Available: https://doi.org/10.1038/nature14236

[9] J. Ibarz, J. Tan, C. Finn, M. Kalakrishnan, P. Pastor, and S. Levine, "How to train your robot with deep reinforcement learning: lessons we have learned," *The International Journal of Robotics Research*, vol. 40, no. 4-5, pp. 698–721, 2021. [Online]. Available: https://doi.org/10.1177/0278364920987859

[10] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. Al Sallab, S. Yogamani, and P. Pérez, "Deep reinforcement learning for autonomous driving: A survey," *IEEE Transactions on Intelligent Transportation Systems*, 2021. doi: 10.1109/TITS.2021.3054625

[11] T. Tajmajer, "Modular multi-objective deep reinforcement learning with decision values," in *2018 Federated conference on computer science and information systems (FedCSIS)*. IEEE, 2018, pp. 85–93. [Online]. Available: http://dx.doi.org/10.15439/2018F231

[12] L. Sun and Y. Gong, "Active learning for image classification: A deep reinforcement learning approach," in *2019 2nd China Symposium on Cognitive Computing and Hybrid Intelligence (CCHI)*. IEEE, 2019. doi: 10.1109/CCHI.2019.8901911 pp. 71–76.

[13] Z. Liu, J. Wang, S. Gong, H. Lu, and D. Tao, "Deep reinforcement active learning for human-in-the-loop person re-identification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019. doi: 10.1109/ICCV.2019.00622 pp. 6122–6131.

[14] V. R. Saripalli, D. Pati, M. Potter, G. Avinash, and C. W. Anderson, "Ai-assisted annotator using reinforcement learning," *SN Computer Science*, vol. 1, no. 6, pp. 1–8, 2020. [Online]. Available: https://doi.org/10.1007/s42979-020-00356-z

[15] J. Wang, Y. Yan, Y. Zhang, G. Cao, M. Yang, and M. K. Ng, "Deep reinforcement active learning for medical image classification," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2020, pp. 33–42. [Online]. Available: https://doi.org/10.1007/978-3-030-59710-8_4

[16] J. Shim, S. Kang, and S. Cho, "Active learning of convolutional neural network for cost-effective wafer map pattern classification," vol. 33, no. 2. IEEE, 2020. doi: 10.1109/TSM.2020.2974867 pp. 258–266.

[17] F.-Q. Liu and Z.-Y. Wang, "Automatic "ground truth" annotation and industrial workpiece dataset generation for deep learning," *International Journal of Automation and Computing*, pp. 1–12, 2020.

[18] H. Liang, L. Yang, H. Cheng, W. Tu, and M. Xu, "Human-in-the-loop reinforcement learning," in *2017 Chinese Automation Congress (CAC)*, 2017. doi: 10.1109/CAC.2017.8243575 pp. 4511–4518.

[19] L. Torrey and M. Taylor, "Teaching on a budget: Agents advising agents in reinforcement learning," in *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, 2013, pp. 1053–1060.

[20] Z. Lin, B. Harrison, A. Keech, and M. O. Riedl, "Explore, exploit or listen: Combining human feedback and policy model to speed up deep reinforcement learning in 3d worlds," *arXiv preprint arXiv:1709.03969*, 2017.

[21] S. Krening, "Humans teaching intelligent agents with verbal instruction," Ph.D. dissertation, Georgia Institute of Technology, 2019.

[22] W. B. Knox and P. Stone, "Tamer: Training an agent manually via evaluative reinforcement," in *2008 7th IEEE International Conference on Development and Learning*. IEEE, 2008, pp. 292–297.

[23] R. Arakawa, S. Kobayashi, Y. Unno, Y. Tsuboi, and S.-i. Maeda, "Dqn-tamer: Human-in-the-loop reinforcement learning with intractable feedback," *arXiv preprint arXiv:1810.11748*, 2018.

[24] G. Li, B. He, R. Gomez, and K. Nakamura, "Interactive reinforcement learning from demonstration and human evaluative feedback," in *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2018. doi: 10.1109/RO-MAN.2018.8525837 pp. 1156–1162.

[25] N. Navidi, "Human ai interaction loop training: New approach for interactive reinforcement learning," *arXiv preprint arXiv:2003.04203*, 2020.

[26] T. Mandel, Y.-E. Liu, E. Brunskill, and Z. Popovic, "Where to add actions in human-in-the-loop reinforcement learning." in *AAAI*, 2017, pp. 2322–2328.

[27] N. Tajbakhsh, L. Jeyaseelan, Q. Li, J. N. Chiang, Z. Wu, and X. Ding, "Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation," *Medical Image Analysis*, p. 101693, 2020. [Online]. Available: https://doi.org/10.1016/j.media.2020.101693

[28] J. C. Caicedo and S. Lazebnik, "Active object localization with deep reinforcement learning," in *Proceedings of the IEEE international conference on computer vision*, 2015. doi: 10.1109/ICCV.2015.286 pp. 2488–2496.

[29] M. Otoofi, “Object localization using deep reinforcement learning Mohammad Otoofi,” Master’s thesis, University of Glasgow, Scotland, 2018.

[30] D. L. Poole and A. K. Mackworth, *Artificial Intelligence: foundations of computational agents*. Cambridge University Press, 2010.

[31] R. Padilla, S. L. Netto, and E. A. da Silva, “A survey on performance metrics for object-detection algorithms,” in *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*. IEEE, 2020. doi: 10.1109/IWSSIP48289.2020.9145130 pp. 237–242.

[32] H. Amin and W. J. Siddiqui, “Cardiomegaly,” *StatPearls [internet]*, 2020.

[33] K. Monowar, “National institutes of health chest x-ray dataset,” May 2020. [Online]. Available: https://www.kaggle.com/khanfashee/nih-chest-x-ray-14-224x224-resized

[34] C. Semsarian, J. Ingles, M. S. Maron, and B. J. Maron, “New perspectives on the prevalence of hypertrophic cardiomyopathy,” *Journal of the American College of Cardiology*, vol. 65, no. 12, pp. 1249–1254, 2015. doi: 10.1016/j.jacc.2015.01.019

[35] B. J. Maron, J. M. Gardin, J. M. Flack, S. S. Gidding, T. T. Kurosaki, and D. E. Bild, “Prevalence of hypertrophic cardiomyopathy in a general population of young adults: echocardiographic analysis of 4111 subjects in the cardia study,” *Circulation*, vol. 92, no. 4, pp. 785–789, 1995. doi: 10.1161/01.cir.92.4.785

[36] M. L. Kwan, L. H. Kushi, E. Weltzien, B. Maring, S. E. Kutner, R. S. Fulton, M. M. Lee, C. B. Ambrosone, and B. J. Caan, “Epidemiology of breast cancer subtypes in two prospective cohort studies of breast cancer survivors,” *Breast Cancer Research*, vol. 11, no. 3, p. R31, 2009. doi: 10.1186/bcr2261

[37] M. Moghbel, C. Y. Ooi, N. Ismail, Y. W. Hau, and N. Memari, “A review of breast boundary and pectoral muscle segmentation methods in computer-aided detection/diagnosis of breast mammography,” *Artificial Intelligence Review*, pp. 1–46, 2019. [Online]. Available: https://doi.org/10.1007/s10462-019-09721-8

[38] V. Gupta, C. Taylor, S. Bonnet, L. M. Prevedello, J. Hawley, R. D. White, M. G. Flores, and B. S. Erdal, “Deep learning-based automatic detection of poorly positioned mammograms to minimize patient return visits for repeat imaging: A real-world application,” *arXiv preprint arXiv:2009.13580*, 2020.

14^{th} Workshop on Computer Aspects of Numerical Algorithms

NUMERICAL algorithms are widely used by scientists engaged in various areas. There is a special need of highly efficient and easy-to-use scalable tools for solving large scale problems. The workshop is devoted to numerical algorithms with the particular attention to the latest scientific trends in this area and to problems related to implementation of libraries of efficient numerical algorithms. The goal of the workshop is meeting of researchers from various institutes and exchanging of their experience, and integrations of scientific centers.

TOPICS

- Parallel numerical algorithms
- Novel data formats for dense and sparse matrices
- Libraries for numerical computations
- Numerical algorithms testing and benchmarking
- Analysis of rounding errors of numerical algorithms
- Languages, tools and environments for programming numerical algorithms
- Numerical algorithms on coprocesors (GPU, Intel Xeon Phi, etc.)
- Paradigms of programming numerical algorithms
- Contemporary computer architectures
- Heterogeneous numerical algorithms
- Applications of numerical algorithms in science and technology

TECHNICAL SESSION CHAIRS

- **Bylina, Beata,** Maria Curie-Sklodowska University, Poland
- **Bylina, Jaroslaw,** Maria Curie-Sklodowska University, Poland
- **Stpiczyński, Przemysław,** Maria Curie-Sklodowska University, Poland

PROGRAM COMMITTEE

- **Amodio, Pierluigi,** Universita' di Bari, Italy
- **Anastassi, Zacharias,** ASPETE School of Pedagogical and Technological Education, United Kingdom
- **Banaś, Krzysztof,** AGH University of Science and Technology, Poland
- **Bielecki, Włodzimierz,** West Pomeranian University of Technology in Szczecin, Poland
- **Brugnano, Luigi,** Università di Firenze, Italy
- **Burczynski, Tadeusz,** Polish Academy of Sciences, Poland
- **Czachórski, Tadeusz,** Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, Gliwice, Poland
- **Domanska, Joanna,** Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, Poland
- **Fialko, Sergiy,** Cracow University of Technology, Poland
- **Gemignani, Luca,** University of Pisa, Italy
- **Gepner, Paweł**
- **Giannoutakis, Konstantinos,** CERTH-ITI, Greece
- **Georgiev, Krassimir,** Bulgarian Academy of Sciences, Institute of Information and Communication Technologies, Bulgaria
- **Gravvanis, George,** Democritus University of Thrace, Greece
- **Kozielski, Stanisław,** Silesian University of Technology, Institute of Informatics, Poland
- **Krawczyk, Henryk,** Gdańsk University of Technology, Poland
- **Kucaba-Pietal, Anna,** Politechnika Rzeszowska, Poland
- **Lirkov, Ivan,** Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, Poland
- **Ltaief, Hatem,** King Abdullah University of Science and Technology, Saudi Arabia
- **Luszczek, Piotr,** University of Tennessee Knoxville, United States
- **Marowka, Ami,** Bar-Ilan University, Isreal
- **Mehmood, Rashid,** King Abdulaziz University, Saudi Arabia
- **Mrozek, Dariusz,** Silesian University of Technology, Institute of Informatics, Poland
- **Palkowski, Marek,** Faculty of Computer Science, West Pomeranian University of Technology, Poland
- **Petcu, Dana,** West University of Timisoara, Romania
- **Rojek, Krzysztof,** Czestochowa University of Technology, Poland
- **Sawerwain, Marek,** University of Zielona Góra, Poland
- **Sidje, Roger B.,** University of Alabama, United States
- **Siminski, Krzysztof,** Silesian University of Technology, Poland
- **Skubalska-Rafajłowicz, Ewa,** Wrocław University of Science and Technology, Poland
- **Trivedi, Kishor S.,** Duke University, United States
- **Tudruj, Marek,** Polish-Japanese Institute of Information Technology, Institute of Computer Science, Polish Academy of Sciences, Poland
- **Ustimenko, Vasyl,** University of Maria Curie Sklodowska in Lublin, Poland

- **Wyrzykowski, Roman,** Czestochowa University of Technology, Poland
- **Vajtersic, Marian,** Deaprtment of Computer Sciences, University of Salzburg, Switzeland
- **Vardaneg, Tullio ,** University of Padova, Italy
- **Vazhenin, Alexander,** University of Aizu, Japan

DOI: 10.15439/2021F68
ISSN 2300-5963 ACSIS, Vol. 25

The impact of vectorization and parallelization of the slope algorithm on performance and energy efficiency on multi-core architecture

Beata Bylina, Joanna Potiopa, Michał Klisowski, Jarosław Bylina
Institute of Computer Science, Marie Curie-Sklodowska University
Pl. M. Curie-Skłodowskiej 5
Lublin, 20-031, Poland
Email: {beata.bylina, joanna.potiopa, michal.klisowski, jaroslaw.bylina}@umcs.pl

***Abstract*—Calculation of land-surface parameters (e.g. slope, aspect, curvature) is an important part of many geospatial analyses. Current research trends are aimed at developing new software techniques to achieve the best performance and energy trade-off. In our work, we concentrate on the vectorization and parallelization to improve overall energy efficiency and performance of the neighborhood raster algorithms for the computation of land-surface parameters. We chose the slope calculation algorithm as the basis for our investigation. The parallelization was achieved through redesigning the the original sequential code with OpenMP SIMD vectorization hints for compiler, OpenMP loop parallelization, and the hybrid of these techniques. To evaluate both performance and energy savings, we tested our vector-parallel implementations on a multi-core computer for various data sizes. RAPL interface was used to measure energy consumption. The results showed that optimization towards high performance can also be an effective strategy for improving energy efficiency.**

***Index Terms*—power, energy efficiency, RAPL, slope, multicore**

I. Introduction

WITH the growing demand for computing power new, more performant computer architectures have emerged. At the same time, the development of computer systems entails an increase in electricity consumption. One way to achieve the reduction of electricity consumption is modern, more energy-efficient hardware. It allows for high computing performance with lower energy consumption. Another way of achieving energy efficiency in high performance computing (HPC) is rethinking the software, including both runtime environments and applications themselves. Efforts are constantly being made to study the impact of algorithm optimizations on energy consumption [11], [6]. Yet, there are still many algorithms formulated in the past that now need to be rewritten to make effective use of modern computer architectures.

Examples are various geospatial analysis algorithms. Many of them are very time-consuming and does not scale well with large data sets, but are applied for increasingly large areas or at increasing resolution. Traditional GIS (geographic information system) software implements sequential algorithms that do not use efficiently computing power of modern computer architectures.

The study of the effect of terrain profile on hydrological, geomorphological, and ecological phenomena and processes start with the calculation of topographic parameters from the digital elevation model (DEM) [4]. The slope is one of the basic primary parameters and it is used e.g. for computing flow velocity for both overland and channelized flow. Other parameters (like soil erosion and deposition, soil wetness, flow speed) are calculated based on the slope. The slope calculation algorithm plays a fundamental role in more advanced models, although it is only one of the input elements for advanced computational algorithms (like modeling rates of snowmelt and evapotranspiration) [10]. Therefore, the high-performance slope calculation algorithm in such complex geospatial analyses is the key to speed them up. Transformations similar to those applied by us to the slope algorithm can be used to other related geospatial algorithms based on the neighborhood relation (e.g. aspect, curvature, focal flow).

It is important to use performant and energy-efficient parallel processing on multi-core machines or hybrid cluster systems for spatial analysis such as slope. Currently, it is not possible to create an architecture-independent solution to the problem of optimization of algorithms in terms of energy efficiency because energy consumption is closely related to a specific architecture. However, some results show that techniques that optimize the performance of algorithms can also improve energy efficiency [11].

In this article, we investigate the impact of the optimization of the slope algorithm on performance, power and energy consumption, and the correlation between them on multicore architecture. We use this architecture because many cores enable running multiple processes at the same time with greater ease, increasing the performance of applications and programs, especially those operating on large data size. The slope algorithm acceleration is achieved by vector optimization for each core and parallel implementation for multi-core processors. The energy efficiency of the proposed solutions is assessed for data (DEM files) of various sizes. The Intel RAPL (Running Average Power Limit) [3], [9] interface was used as a source of information on energy consumption.

The main contributions of this article are following:

- Results of the tests and conclusions from the evaluation of the execution time and acceleration of different versions of the slope algorithm for various data sizes.
- Conclusions on the impact of the optimization techniques on power consumption.
- Conclusions on the impact of the effect of vectorization and parallelization on energy consumption.
- Analysis of the correlation between performance and energy consumption.

This paper is organized as follows. Section 2 presents related works. Section 3 is devoted to the slope algorithm. It explains what slope is and describes the algorithm used to calculate it. It also describes versions of the algorithm tuned to the architecture used: version that enables the use of vector registers and two parallel versions. In Section 4, we concentrate on the details of conducting tests and on the discussion and explanation of the results. In Section 5, we present the conclusions of the conducted experiment and further research directions.

II. Related works

Many studies are targeted at the prediction of power consumption and energy savings for various processing units. These issues can be addressed at both the hardware and software levels. The software-level approach involves redesigning numerical algorithms from various fields in terms of energy efficiency. Some examples one can find in [7], [6], [8], [2], [1] and [11]. In this, paper we also take the software-level approach. We focus on general-purpose processors (CPUs) with vector units and selected geospatial algorithms.

To take full advantage of the computing potential of CPUs while ensuring energy efficiency, both vector and multicore processing should be used. The works [7], [6], and [8] showed advantages and limitations of vectorization for energy efficiency. They describe the techniques of automatic and manual vectorization of Gauss elimination and Gram-Smith orthogonalization algorithms for multicore computers. In our work, we investigate the energy efficiency of the code manually vectorized with compiler directives. The directives inform the compiler about the vectorizable instructions in an appropriately transformed code.

In [2] and [1], the authors study energy efficiency in the context of high-performance dense linear algebra libraries for multicore computers and multithreading. Both works used transformed block matrix decomposition algorithms. Energy consumption measurements were reported along with parallel performance numbers on multi-socket machines. The conclusion was that the use of block linear algebra algorithms results in energy savings. In our work, we study the energy efficiency of algorithms employing loop fission transformations.

The paper [11] examines the impact of performance optimization on the power and energy consumption of Intel Xeon Scalable processors. The studies were conducted on the example of the MPDATA application (a finite-difference solver for geophysical flows). MPDATA is a memory-bound application and it needed optimization to utilize both vector and multicore processing. The research showed that improving memory access (i.e. cache reusing and data locality) for such memory-bound applications also improves energy efficiency. Additionally, the authors show that SIMD vectorization can lead to energy consumption reduction and, at the same time, increase the efficiency of calculations. They also evaluate the CPU frequency scaling as a tool for balancing energy savings with admissible performance losses.

In our research, we investigate the effect of vectorization and combining it with multithreading on energy efficiency for geospatial raster algorithms on a multi-core machine with vector units.

III. HPC Slope Algorithm

Digital elevation model (DEM) is a digital representation of earth's surface. DEM and its derivatives (land-surface parameters) are the basis of various geomorphometric analyses [4]. The slope is one of the most important and most frequently computed land-surface parameters.

DEM is most frequently represented as a two-dimensional regular grid of cells. Each cell contains an elevation value. The same representation is used for land-surface parameters, such as the slope.

All the algorithms discussed in this section use this representation for input and output data.

Various ways of calculating slope from DEM are discussed in [12] and [13]. Our implementation uses the method described in [5]. This method is also implemented in most popular GIS software packages (ArcGIS, QGIS, GRASS GIS, SAGA GIS).

In this section, we discuss some details of our implementation and its improvements, i.e. vectorization and parallelization.

A. Basic algorithm

Slope at a given point can be described as the maximum rate of change of elevation value at that point. It can be expressed as a slope angle (between 0 and 90 degrees). The method of determining the slope is presented by Algorithm 1. The algorithm requires two arrays: dem stores elevation values, and $slope$ — computed slope values in each cell. dem is an input array, $slope$ is an output array. These arrays have got the same dimensions. Additionally, we also need information about cell size (Δx, Δy) and the number of rows n and columns m of arrays. Δx means the grid interval from west to east, expressed in the same units as the elevation in dem. Δy means the grid interval from south to north, expressed in the same units as the elevation. In the basic implementation of slope, all instructions are executed sequentially, one by one.

B. Vectorization

The slope algorithm reads its input data from the main memory. It also writes its output to the main memory. This results in intensive data movement to and from the main memory. This movement severely limits the performance. To prevent this and make better use of cache we developed a

Algorithm 1: Base: Basic sequential algorithm

```
Input: $dem$ — input DEM
$\Delta x$ — west-to-east cell size
$\Delta y$ — south-to-north cell size
$n$ — number of rows of $dem$
$m$ — number of columns of $dem$
Output: $slope$ — output of the same size as $dem$
for $r \leftarrow 1 \ldots (n-2)$ do
    for $c \leftarrow 1 \ldots (m-2)$ do
        $p \leftarrow ((dem[r-1][c+1] + 2dem[r][c+1]$
            $+ dem[r+1][c+1])$
            $- (dem[r-1][c-1] + 2dem[r][c-1]$
            $+ dem[r+1][c-1]))$
            $/(8\Delta x)$
        $q \leftarrow ((dem[r+1][c-1] + 2dem[r+1][c]$
            $+ dem[r+1][c+1])$
            $- (dem[r-1][c-1] + 2dem[r-1][c]$
            $+ dem[r-1][c+1]))$
            $/(8\Delta y)$
        $slope[r][c] \leftarrow \arctan(\sqrt{p^2+q^2})$
return $slope$
```

new version of the algorithm. The modification consisted in transforming the inner loop. We use the loop fission technique [14]. Algorithm 2 describes the transformed slope algorithm. The modified version improves cache usage, reduces main memory access, and enables the use of vector registers found in every modern CPU. It employs the vectorization along the c-dimension using the **#pragma omp simd** directive from the OpenMP standard.

C. Parallelization

Algorithms 3 and 4 describe parallelized algorithms 1 and 2 (base and transformed). In both parallel implementations, the outermost loop is processed in parallel using the OpenMP standard. These versions make use of data parallelism in r-dimension with the **#pragma omp parallel for** directive.

IV. Numerical experiment – methodology and results analysis

A. Methodology

We benchmark four versions of the slope algorithm:

- Base (Algorithm 1) – basic sequential algorithm,
- Parallelized base (Algorithm 3) – parallel version of Base,
- Transformed with SIMD (Algorithm 2) – transformed sequential algorithm with vectorization hints for compiler,
- Parallelized transformed with SIMD (Algorithm 4) – parallel version of Transformed with SIMD; The code is transformed so that it can be executed in parallel on multiple cores and that it can be efficiently vectorized.

All versions have been implemented in C++.

Algorithm 2: Transformed with SIMD: Transformed sequential algorithm

```
Input: $dem$ — input DEM
$\Delta x$ — west-to-east cell size
$\Delta y$ — south-to-north cell size
$n$ — number of rows of $dem$
$m$ — number of columns of $dem$
Output: $slope$ — output of the same size as $dem$
for $r \leftarrow 1 \ldots (n-2)$ do
    /* calculating $p$ */
    #pragma omp simd
    for $c \leftarrow 1 \ldots (m-2)$ do
        $p[c] \leftarrow dem[r-1][c+1] - dem[r-1][c-1]$
    #pragma omp simd
    for $c \leftarrow 1 \ldots (m-2)$ do
        $p[c] \leftarrow p[c]$
            $+ 2(dem[r][c+1] - dem[r][c-1])$
    #pragma omp simd
    for $c \leftarrow 1 \ldots (m-2)$ do
        $p[c] \leftarrow (p[c] + (dem[r+1][c+1]$
                $- dem[r+1][c-1]))/(8\Delta x)$
    /* calculating $q$ */
    #pragma omp simd
    for $c \leftarrow 1 \ldots (m-2)$ do
        $q[c] \leftarrow dem[r+1][c-1]$
            $+ 2dem[r+1][c] + dem[r+1][c+1]$
    #pragma omp simd
    for $c \leftarrow 1 \ldots (m-2)$ do
        $q[c] \leftarrow (q[c] - (dem[r-1][c-1]$
                $+ 2dem[r-1][c]$
                $+ dem[r-1][c+1]))/(8\Delta y)$
    #pragma omp simd
    /* calculating $slope$ */
    for $c \leftarrow 1 \ldots (m-2)$ do
        $slope[r][c] \leftarrow \arctan(\sqrt{p[c]^2 + q[c]^2})$
return $slope$
```

For tests, we used 1-meter resolution GeoTIFF files. The area of the smallest one we denote by R. We consider GeoTIFF files in 10 different sizes: from R to 10R, covering areas from 200 km^2 to 2000 km^2. The use of multiples of R facilitates analysis of scalability in data size. The sizes of the test data are presented in Table I. Each test area is an $m \times n$ matrix (m columns, n rows). The data type of each element of input and output arrays, as well as the type used for all calculations, is a single-precision floating-point type (4 bytes).

The performance and power measurements presented in this work were performed on a computing platform equipped with a modern multi-core processor code-named Haswell and with the following parameters:

Algorithm 3: Parallelized base: Parallelized basic algorithm

```
Input: $dem$ — input DEM
$\Delta x$ — west-to-east cell size
$\Delta y$ — south-to-north cell size
$n$ — number of rows of $dem$
$m$ — number of columns of $dem$
Output: $slope$ — output of the same size as $dem$
#pragma omp parallel for
for $r \leftarrow 1 \ldots (n-2)$ do
    for $c \leftarrow 1 \ldots (m-2)$ do
        ...
        /* loop body as in Alg. 1 */
        ...
return $slope$
```

Algorithm 4: Parallelized transformed with SIMD: Transformed sequential algorithm

```
Input: $dem$ — input DEM
$\Delta x$ — west-to-east cell size
$\Delta y$ — south-to-north cell size
$n$ — number of rows of $dem$
$m$ — number of columns of $dem$
Output: $slope$ — output of the same size as in $dem$
#pragma omp parallel for
for $r \leftarrow 1 \ldots (n-2)$ do
    ...
    /* loop body as in Alg. 2 */
    ...
return $slope$
```

```
processor: 2x Intel Xeon E5-2670 v3 @ 2.30GHz
(2x12 cores with HT)
RAM: 128GB (8x16GB DDR4 2133MHz ECC)
```

The machine is equipped with 2 processors 12 cores each. Thus the program can be run in 24 threads (the number of threads equals the number of cores). In addition, this processor supports Intel Hyper-Threading (HT) technology and allows concurrent execution of 2 threads on one processor core. However, we do not use it during our tests.

The following software was installed during tests:

```
operating system: CentOS 7.6
kernel: Linux 3.10.0
GCC: 8.3.1 z OpenMP 4.5
GDAL: 2.4.0
```

The programs were compiled with the GCC compiler and the optimization flag `-O3` turned on. The GDAL (Geospatial Data Abstraction Library) [3] library was used to read from and write to GeoTIFF files. In each version of the algorithm, to help the optimization, data alignment in memory was used.

TABLE I: Characteristics of test areas

Area	Number of rows (n)	Number of columns (m)	Number of cells ($m \times n$)	[GB]
R	4000	50000	200000000	0.75
2R	8000	50000	400000000	1.49
3R	12000	50000	600000000	2.24
4R	16000	50000	800000000	2.98
5R	20000	50000	1000000000	3.73
6R	24000	50000	1200000000	4.47
7R	28000	50000	1400000000	5.22
8R	32000	50000	1600000000	5.96
9R	36000	50000	1800000000	6.71
10R	40000	50000	2000000000	7.45

In this chapter, we also describe the impact of the algorithm modifications on power and energy consumption. The measurements were performed using the Intel's Running Average Power Limit (RAPL) interface, which is available for all Intel processors, starting with the Sandy Bridge architecture. RAPL uses machine-specific registers to monitor and control power consumption in real-time. On multi-socket systems, RAPL provides the results for each socket (each package) separately. RAPL also provides separate measurement values for the memory modules (DRAM) associated with each socket. Starting from Haswell processors which are equipped with fully integrated voltage regulators, the accuracy of the measurements returned by RAPL has significantly improved [9].

B. Execution Time

Each of the implemented algorithms requires reading the input data and writing the results. The times of these operations do not differ between algorithms for the same data size. Table 1 shows the average time of reading and writing data for each input data size.

TABLE II: Average read and write time [s] for each input data size

Area	Reading [s]	Writing [s]
R	0.84	4.04
2R	1.51	8.45
3R	2.24	11.50
4R	2.81	15.56
5R	4.65	20.63
6R	4.58	24.41
7R	5.98	30.17
8R	5.66	32.52
9R	7.40	37.99
10R	7.08	40.37

First, we measured the execution time of each algorithm for different sizes of input data. Execution times are given for calculations only — without reading and writing data. The results are presented in Figure 1. Tests show that the transformation of the algorithm accompanied with vectorization hints shortens the execution time by about 30% compared to the base version. The use of parallelism reduces the execution time much more: parallelization alone reduces the time by more than 80% compared to the base version, while the parallelized and vectorized version shortens execution time by almost 90% for each test data size.

Table III shows the speedup gained by vectorization and parallelization for selected data sizes. In table III, the pa-

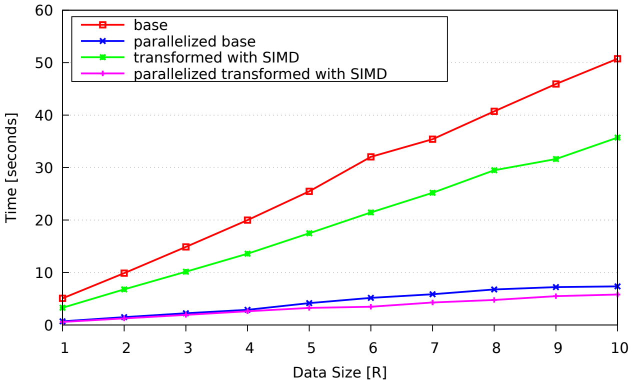


Fig. 1: Execution time of slope algorithms for different data sizes

TABLE III: Relative speedup of the slope algorithms

	R	5R	10R
T_{Base} / $T_{Transformed}$	1.55	1.46	1.42
$T_{ParallelizedBase}$ / $T_{ParallelizedTransformed}$	1.22	1.29	1.27
T_{Base} / $T_{ParallelizedBase}$	7.33	6.11	6.90
$T_{Transformed}$ / $T_{ParallelizedTransformed}$	5.81	5.40	6.16
T_{Base} / $T_{ParallelizedTransformed}$	8.98	7.87	8.76

rameter $T_{Algorithm}$ denotes the execution time of the given algorithm. The table shows that the vectorization of the base version gives a greater improvement (from 1.42 to 1.55) compared to the vectorization of the parallelized version (from 1.22 to 1.29). The very parallelization of the base version of the algorithm speeds up the execution about 7 times, while the program using OpenMP pragmas and the use of SIMD extensions (Parallelized Transformed with SIMD) is performed on average 8 times faster and for some sizes even 9 times faster.

C. Evaluation of Power and Energy Consumption

1) Power Consumption for Different Slope Versions: Next, we evaluate the impact of the applied performance optimization steps on the power consumption, which is measured with the RAPL interface. RAPL counters are updated once every 1 ms and have an adjustable sampling rate that has been set to 100 ms.

Figure 2 shows the power profiles of the Base algorithm implementation for three data sizes: R, 5R, 10R. The experiment was conducted on a dual-socket system, therefore RAPL returns the measurement results separately for packages attached to each socket: Package0 and Package1, and separately for the memory attached to the package's integrated memory controller: DRAM0 (attached to Package0) and DRAM1 (Package1). Figure 2 shows power profiles for the execution of the Base algorithm successively for sizes R, 5R, and 10R. Between the runs, the system is idle for 3 seconds. The Base algorithm is sequential, so only one core from the selected socket is being employed at any time. The figure shows the power consumption of the system components during program execution for the data size R (3–15 sec.), 5R (17–71 sec.), and 10R (73–174 sec.), respectively. Data reading times (R_{read}, $5R_{read}$, $10R_{read}$ — marked in green), computation times (R_{comp}, $5R_{comp}$, $10R_{comp}$ — marked in red), and results writing times (R_{write}, $5R_{write}$, $10R_{write}$ — marked in blue) are shown above the power consumption graphs.

Figure 3 shows the power profiles of the execution of four versions of the slope algorithm: Base, Transformed with SIMD, Parallelized, and Parallelized transformed with SIMD. Sequential algorithms (Base and Transformed with SIMD) use exactly one core out of 24 available. Parallel versions of algorithms (Parallelized base and Parallelized transformed with SIMD) utilize all 24 available cores. As you can see in Figures 3a–3d, the system is idle before and after the execution of a program. In this state, it consumes about 39 W (about 20 W for each socket: Package + DRAM). This is the minimum power necessary to keep the system idle. After the start of the program, more system components are involved (cores, memory), which increases the power consumption. The system returns to the idle state when it completes the program execution. One can observe that for modified versions (Transformed with SIMD, Parallelized, Parallelized transformed with SIMD), the system returns to the idle state faster (the program execution takes less time), and because the reading and writing time is constant for a given size, the calculation time itself is shorter. Graphs 3a and 3b show that the levels of power consumption for the sequential versions of the algorithm are close to each other. The maximum instantaneous power of the entire system (both sockets: Package0 + DRAM0, Package1 + DRAM1) is 83 W for the Base version and 82 W for the Transformed with SIMD version. Graphs 3c and 3d show the power consumption for the parallel versions. They show that only when calculations are performed, the power consumption increases and the cores on both sockets work simultaneously. The maximum instantaneous power for the Parallelized base version is 190 W, and for the Parallelized transformed with SIMD version it is 219 W. Reading and writing data does not differ between versions (and these are operations performed sequentially), so the power consumption for these operations is the same for all algorithms.

Regardless of the version of the algorithm, reading and writing data takes the same time and power, and thus when analyzing the average power and energy consumption, we consider the computation alone.

Figure 4 shows a comparison of the average power consumption of the considered system components (Package0 + DRAM0, Package1 + DRAM1) by different versions of the slope algorithm and for different data sizes. It also shows the value of the system idle power. The graph shows that for the sequential versions (Base, Transformed with SIMD) the average power consumption is almost identical for each size. Parallel versions of algorithms run faster (Figure 1) but have higher average power consumption. The average power consumption of the Parallel base version is on average 43% higher than the Base version, and the average power consumption of the Parallelized transformed with SIMD version is on average 31% higher than the Base version. Despite having the highest

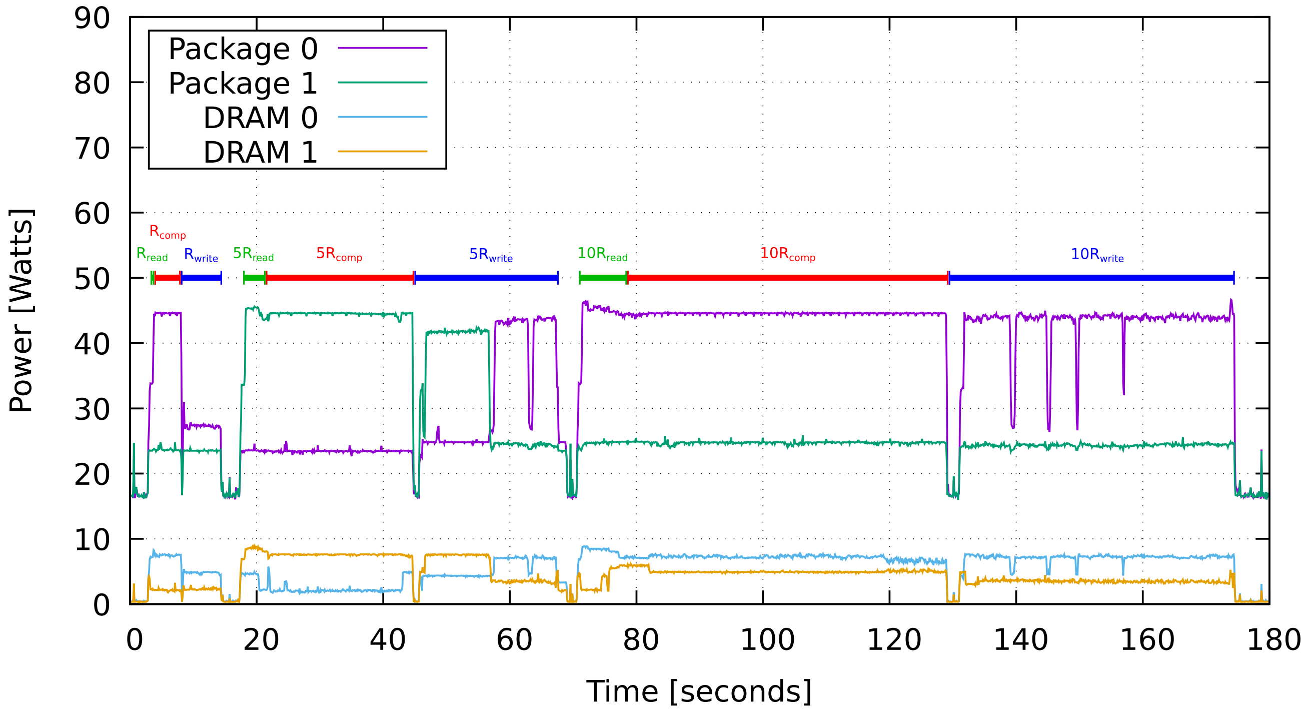


Fig. 2: The power profiling of Base algorithm for R, 5R, 10R with RAPL

instantaneous power, the Parallelized transformed with SIMD version has a lower (2–16%) average power consumption than the Parallelized base version.

2) Energy consumption: In this section, we show how the performance optimization steps affect the total computation energy consumption.

Table IV presents a comparison of energy and performance-related parameters for different versions of the slope algorithm for a data size of 10R. In addition to the execution time, the total energy consumption obtained from the RAPL interface is given. Moreover, the performance (in Mflop/s) and the energy efficiency (in Mflop/J) are given. Energy efficiency is expressed as the ratio of the number of floating-point operations to the total energy consumption.

Figure 5 shows the effect of different optimization mechanisms on the total energy consumption for different data sizes. One can see that the power consumption increases as the data size increases.

The modification used in the Transformed with SIMD algorithm, which removes the data dependency, allows for more efficient access to memory and the use of vectorization mechanisms, which results in a reduction of the computation time (Figure 1), but also a reduction in energy consumption (27% to 35%, 32% on average). This optimization affects neither the maximum instantaneous power consumption (Figure 3) nor the average power consumption (Figure 4).

The optimization used in the Parallelized base version, i.e. the use of OpenMP pragmas, allows all available cores in the system to work simultaneously. This causes momentary increases in power consumption (Fig. 3) and higher average power consumption (Fig. 4). However, it also results in a significant reduction of the computation time (Fig. 1) and energy consumption (74%–80%, 77% on average).

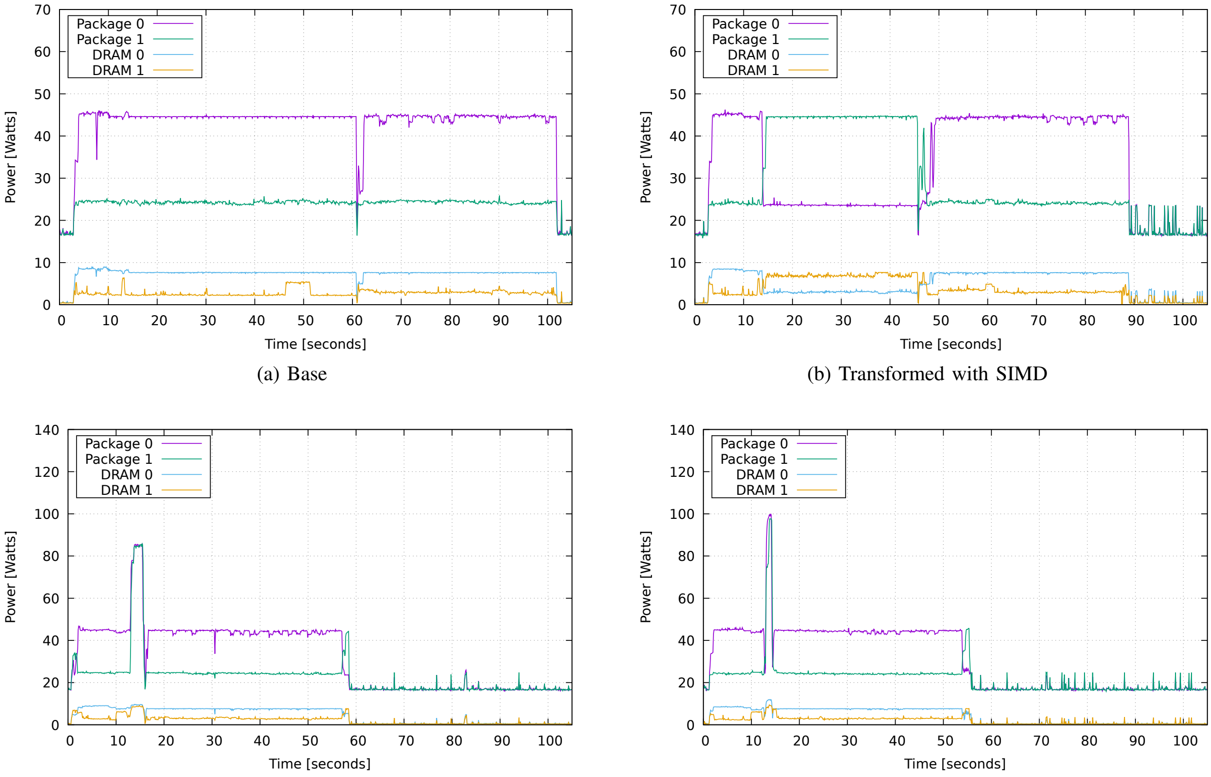
In the Parallelized transformed with SIMD version, the data dependencies are removed and vectorization is introduced into the parallel algorithm. Compared to Parallelized base, energy consumption is reduced by 28%. We also obtain even shorter computation times and lower average power consumption, with higher maximum instantaneous power.

However, the use of both parallelization and vectorization in the Parallelized transformed with SIMD version, reduces the execution time by almost 90% compared to the Base version (Figure 11). Despite the high maximum instantaneous power and higher average power consumption, we reduce energy consumption by an average of 84% compared to the Base version.

V. Conclusion

This article investigates four versions of the slope algorithm. Their execution time, power and energy consumption, and the correlation between performance and energy consumption are discussed. Measurements were made on a dual-socket machine with Intel Xeon E5-2670 processors using the Intel RAPL interface.

Both of the proposed transformations — vectorization and parallelization — reduce the computation time. The results



(a) Base

(b) Transformed with SIMD

(c) Parallelized base

(d) Parallelized transformed with SIMD

Fig. 3: The power profiling of Base, Transformed with SIMD, Parallelized base, Parallelized transformed with SIMD for 10R with RAPL

TABLE IV: Energy efficiency for various slope versions (10R)

Version	Time [s]	Total energy [J]	Performance [Mflop/s]	Energy efficiency [Mflop/J]
base	50.76	4028.18	788.00	0.20
parallelized base	7.35	867.68	5438.96	6.27
transformed with SIMD	35.72	2792.00	1119.58	0.40
parallelized transformed with SIMD	5.80	633.69	6901.30	10.89

show that the most performant version (parallel with vectorization) can shorten the computation time by more than 8 times. Through the analysis of power and energy consumption, one can see that vectorization alone, can slightly speed up the algorithm, without increasing average power consumption or maximum instantaneous power consumption. The reduction in the computation time allows for the reduction of energy consumption by about 30%.

The parallel version enables the simultaneous operation of all system components. This results in a significant reduction in computing time compared to the basic version (by almost 90%), but also an increase in both the maximum (more than 2 times) and average power consumption (by 30%–40%). Finally, however, we achieve a reduction in energy consumption by an average of 84% compared to the Base version. We can see that short periods of increased instantaneous power do not negatively affect the total energy consumption as long as the program computation time is shortened.

The conducted tests show that the proposed solutions respond well to increasing the size of the problem. For the largest data size tested, the energy efficiency improves (from 0.2 Mflop/J for the basic version to 10.8 Mflop/J for the most optimized version) along with the increase in performance.

The slope algorithm is not computationally intensive, but it is one of the basic components used in other geomorphometric analyses. The increase of performance that also causes the reduction of the energy consumption will improve the energy efficiency of the secondary analyses. Moreover, the proposed transformations can be applied to other raster analyses employing similar techniques (neighborhood relation) such as computation of aspect, curvature, and flow direction.

Seeing the improvement in energy efficiency after adapting the slope algorithm to a multi-core system, we plan to investigate the impact of the optimization methods applied to similar algorithms on the latest Ice Lake Intel Xeon processors and on the new generation of AMD Rome EPYC processors. We

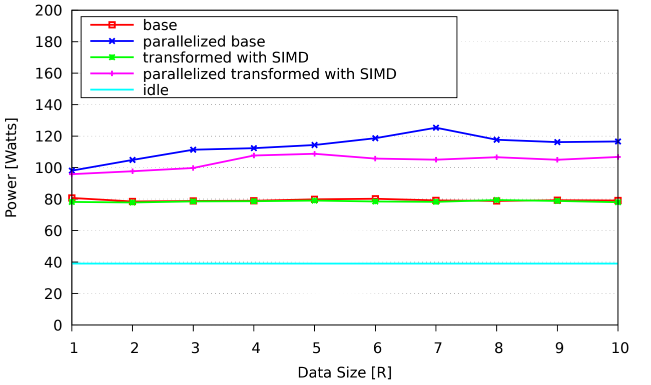


Fig. 4: Average power consumption as a function of different data sizes — and the idle power level

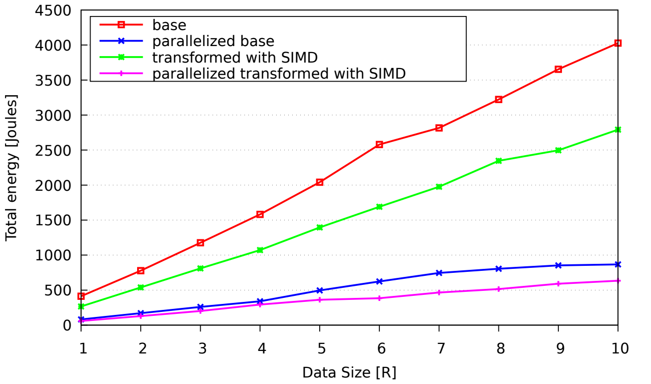


Fig. 5: Total energy as a function of different data sizes

also plan to investigate and improve other geospatial raster algorithms in terms of performance and energy efficiency.

References

[1] B. Bylina and J. Bylina. Studying OpenMP thread mapping for parallel linear algebra kernels on multicore system. *Bulletin of the Polish Academy of Sciences*, 66(6):981–990, 2018.

[2] J. Dongarra, H. Ltaief, P. Luszczek, and V. M. Weaver. Energy footprint of advanced dense numerical linear algebra using tile algorithms on multicore architectures. In *2012 Second International Conference on Cloud and Green Computing*, pages 274–281, 2012.

[3] D. Hackenberg, R. Schöne, T. Ilsche, D. Molka, J. Schuchart, and R. Geyer. An energy efficiency feature survey of the Intel Haswell processor. In *2015 IEEE International Parallel and Distributed Processing Symposium Workshop*, pages 896–904, 2015.

[4] T. Hengl and H.I. Reuter, editors. *Geomorphometry: Concepts, Software, Applications*, volume 33. Elsevier, Amsterdam, 2008.

[5] B. K. P. Horn. Hill shading and the reflectance map. *Proceedings of the IEEE*, 69(1):14–47, Jan 1981.

[6] T. Jakobs, B. Naumann, and G. Rünger. Performance and energy consumption of the SIMD Gram–Schmidt process for vector orthogonalization. *The Journal of Supercomputing*, 76:1999–2021, 2019.

[7] T. Jakobs and G. Rünger. Examining energy efficiency of vectorization techniques using a Gaussian elimination. In *2018 International Conference on High Performance Computing Simulation (HPCS)*, pages 268–275, 2018.

[8] T. Jakobs and G. Rünger. On the energy consumption of load/store AVX instructions. In *2018 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 319–327, 2018.

[9] K. Khan, M. Hirki, T. Niemi, J. Nurminen, and Z. Ou. RAPL in action: Experiences in using RAPL for power measurements. *ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS)*, 3, 01 2018.

[10] S.D. Peckham. Chapter 25 Geomorphometry and Spatial Hydrologic Modelling. In Tomislav Hengl and Hannes I. Reuter, editors, *Geomorphometry*, volume 33 of *Developments in Soil Science*, pages 579 – 602. Elsevier, 2009.

[11] L. Szustak, R. Wyrzykowski, T. Olas, and V. Mele. Correlation of performance optimizations and energy consumption for stencil-based application on Intel Xeon scalable processors. *IEEE Transactions on Parallel and Distributed Systems*, 31(11):2582–2593, 2020.

[12] J. Tang, P. Pilesjö, and A. Persson. Estimating slope from raster data – a test of eight algorithms at different resolutions in flat and steep terrain. *Geodesy and Cartography*, 39(2):41–52, 2013.

[13] S. Warren, M. Hohmann, K. Auerswald, and H. Mitasova. An evaluation of methods to determine slope using digital elevation data. *Catena*, pages 215–233, 12 2004.

[14] M. E. Wolf and M. S. Lam. A loop transformation theory and an algorithm to maximize parallelism. *IEEE Transactions on Parallel and Distributed Systems*, 2(4):452–471, 1991.

DOI: 10.15439/2021F38
ISSN 2300-5963 ACSIS, Vol. 25

Bernoulli Meets PBFT: Modeling BFT Protocols in the Presence of Dynamic Failures

Martin Nischwitz, Marko Esche
Physikalisch-Technische Bundesanstalt
Berlin, Germany
Email: {martin.nischwitz, marko.esche}@ptb.de

Florian Tschorsch
Department for Distributed Security Infrastructures
Technische Universität Berlin
Berlin, Germany
Email: florian.tschorsch@tu-berlin.de

***Abstract*—The publication of the pivotal state machine replication protocol PBFT laid the foundation for a body of BFT protocols. We introduce a probabilistic model for evaluating BFT protocols in the presence of dynamic link and crash failures. The model is derived from the communication pattern, facilitating an adaptation to other protocols. The state of replicas is captured and used to derive the success probability of the protocol execution. To this end, we examine the influence of link and crash failure rates as well as the number of replicas. A comparison in protocol behavior of PBFT, Zyzzyva and SBFT is performed.**

I. Introduction

THE rapidly increasing connectivity of devices, as for example envisioned by the Internet of things, entices the development of large-scale, globally distributed systems. The European Metrology Cloud project [1], which aims to coordinate the digital transformation of legal metrology, is a prime example. The scale and complexity of such systems leads to a higher risk of failure and/or malicious behavior. The demand for trust and reliability, however, remains unchanged.

A technique to offer higher fault tolerance and availability for distributed systems is state machine replication (SMR). It requires processes to find agreement on the order of state transitions and, thus, consensus on the system state. One of the most prominent Byzantine fault tolerant (BFT) protocols is Practical Byzantine Fault Tolerance (PBFT) [2]. Many modern systems, including the recent surge of blockchain applications [3], utilize PBFT, or a variation of it, as their core consensus algorithm, e.g., BFT-SMaRt [4], Tendermint [5], RBFT [6], CheapBFT [7], and Hyperledger Fabric v0.6 [8].

While many advanced BFT protocols exist, the impact of dynamic failures in general and unreliable links in particular is often ignored. Many protocols, however, require that messages arrive within a defined timespan, i.e., there is a bound on the message delay. If that bound is not met, the performance of the protocols might deteriorate. To the best of our knowledge, there is unfortunately no technique to assess the impact of unreliable links on the performance of BFT protocols without requiring a comprehensive implementation. Instead, benchmarks on either real systems or simulations are deployed.

In this paper, we fill the gap and present a probabilistic modeling approach for BFT protocols to measure the impact of dynamic link and crash failures on their performance. The model is derived from the communication pattern and therefore transferable to many BFT protocols. It predicts the system state assuming the so-called dynamic link failure model [9], that is, unreliable communication links with message losses and high delays. More specifically, we assume a constant failure probability for all links and processes, model state transitions as Bernoulli trials, and express the resulting system state as probability density functions. Thus, our model can provide feedback already during the design and development phase of BFT protocols as well as support to parameterize timeouts.

Our model validation confirms that the model accurately predicts the probability for successful protocol executions of PBFT as well as BFT-SMaRt [4]. Moreover, we employ our model to Zyzzyva [10], and SBFT [11] to showcase its applicability to other BFT protocols. In our evaluation, we analyze the mentioned protocols, most notably PBFT, with respect to the impact of various failures and protocol stability. Accordingly, the paper's contributions can be summarized as follows:

- We develop a probabilistic model for PBFT to quantify the performance impact in the presence of dynamic link and crash failures. Since the model is based on communication patterns, it is implementation independent.
- We generalize our modeling approach and show that it can be applied to other BFT protocols.
- We validate the approach in a study by comparing it to a simulation of PBFT and BFT-SMaRt and apply it to Zyzzyva and SBFT.
- We identify critical values for dynamic link failure and crash failure rates at which the previously mentioned protocols become unstable.

The remainder of the paper is organized as follows. In Section II, we discuss related work with a focus on BFT modeling and failures. In Section III, we define the system model. Next, we describe the detailed derivation of our modeling approach for PBFT in Section IV, and present a simulation-based model validation in Section V. In Section VI, we use our model to reveal structural differences between PBFT, BFT-SMaRt, Zyzzyva, and SBFT, before we conclude the paper in Section VII.

IEEE Catalog Number: CFP2185N-ART ©2021, PTI

II. Related Work

A. Preliminaries

The main properties of BFT protocols are described by the notions of *safety* and *liveness* [2]. Safety indicates that the protocol satisfies serializability, i.e., it behaves like a centralized system. Liveness, on the other hand, indicates that the system will eventually respond to requests. In order to tolerate f faulty processes, at least $3f + 1$ processes are necessary [12]. Aside from process-related failures, the network, i.e., the communication between processes, also impacts the performance of BFT protocols and is often overlooked.

The network can be described as either synchronous or asynchronous. To bridge the gap between completely synchronous/asynchronous systems, the term *partially synchronous* was introduced [13]. A partially synchronous system may start in an asynchronous state but will, after some unspecified time, eventually return to a synchronous state. This captures temporary link failures, for example. A different perspective on a partially synchronous system is to assume a network with fixed upper bounds on message delays and processing times, where both are unknown a priori. The partially synchronous system model is utilized by many BFT protocols, e.g., PBFT, to circumvent the FLP impossibility [14] and guarantee liveness during the synchronous states of the system, without requiring it at all times. Deterministic BFT protocols guarantee safety, even in the asynchronous state, but require synchronous periods to guarantee liveness.

To detect Byzantine behavior, most BFT protocols utilize timeouts (and signatures). If the happy path of a protocol fails to make progress, a sub-protocol, e.g., a view change protocol, is triggered to recover [15]. To optimize performance, the timeout values should depend on the bounded message delay in the synchronous periods of the network, which plays an important role for deployments [16]. In addition to message delay characteristics, some networks, e.g., wireless networks, might be susceptible to link failures, leading to message omissions or corruptions. These failures are formally captured by the so-called *dynamic link failure model* [9], where the authors prove that consensus is impossible in a synchronous system with an unbounded number of transmission failures. Schmid et al. [17], [18] introduced a hybrid failure model to capture process and communication failures and derived bounds on the number of failures for synchronous networks.

B. BFT Models

Other modeling techniques to analyze the performance of fault tolerant systems have previously been proposed in the literature. The framework HyPerf [19] combines model checking and simulation techniques to explore the possible paths in BFT protocols. While model checking usually proves correctness, their framework uses simulations to explore the possible paths in the model checker and evaluate the performance of the protocol. The model is validated against an implementation of PBFT to predict latencies and throughput.

A method to model PBFT with Stochastic Reward Nets (SRNs) was proposed in [20]. The authors deployed Hyperledger Fabric v0.6, which implements PBFT, and evaluate the mean time to consensus against the number of nodes in the system.

Singh et al. [21] provided the simulation framework BFT-Sim to evaluate BFT protocols. It builds upon the high-level declarative language P2 to implement three different BFT protocols [2], [10], [22] as well as ns-2, to explore various network conditions.

While the previously listed works offer the possibility to evaluate BFT protocols, they all require *comprehensive implementations* of the respective protocol. The model presented in this paper, however, is derived from the *communication pattern*. Moreover, no simulations or measurements are required to employ our model; all system states can be evaluated with closed-form expressions at low computational cost. Finally, the main focus of our work is to present a model for fault tolerant protocols that captures the impact of unreliable communication and varying message delays, which the other models only considered as a minor aspect.

C. Link and Crash Failures

Fathollahnejad et al. [23] examined the impact of link failures on their leader election algorithm in a traffic control system to predict the probability for disagreement, based on Bernoulli trials. As in our paper, the number of received messages of an all-to-all broadcast is modeled with Bernoulli trials. Their protocol, however, does not require the consecutive collection of quorums which is implemented in most fault tolerant (FT) protocols and thus the main focus of the model presented in this paper.

Xu et al. presented RATCHETA [24], a consensus protocol which was designed for embedded devices in a wireless network that might be prone to dynamic link failures. They included an evaluation with artificially induced packet losses, measuring the number of failing consensus instances. RATCHETA requires a trusted subsystem that prevents a process from casting differing votes during the same consensus instance, eliminating the possibility of equivocations. It therefore yields a $2f + 1$ resilience, allowing f Byzantine failures.

In addition, there is a body of literature that covers the theoretical limits of failures of consensus protocols [18], [25], [26], which are based on a hybrid failure model [17] and therefore also capture link failures. Existing models, however, rarely consider the actual impact of unreliable network conditions, such as dynamic crash and link failures, on the protocol algorithm. Since all BFT protocols have a built-in protocol to recover from crashed processes, e.g., view changes, their impact on the performance is tied to the frequency of the recovery algorithm execution.

III. System Model

A. Process Model

The distributed system consists of a fixed number of n processes (we use the term process, node, and replica interchangeably). Typically, no more than f processes are allowed to be subject to Byzantine faults and $n \geq 3f + 1$ replicas are required to guarantee safety [12].

To tolerate Byzantine (or crash) failures in an asynchronous setting, distributed systems rely on timeouts in combination with message thresholds to make progress. Consequently, it is common to describe the protocols in phases, i.e., system states in which each process, e.g., awaits the reception of a certain amount of messages or, alternatively, a timeout. Our model is time-free in that all events are mapped to the respective phases of the protocol.

In order to capture diverse failure cases, e.g., congestion due to high traffic load, we introduce the term *dynamic crash failures*, along the lines of the *dynamic link failure* model by Santoro et al. [9]. That is, processes can become unavailable in each phase. It is assumed that every crashed process will recover almost immediately, upholding its pre-crash state, and may thus be available in the next phase of the protocol. Since most BFT protocols (including PBFT) are based on consecutive phases, a crashed replica will remain inactive until the protocol-specific recovery algorithm, e.g., view-change protocol for PBFT, has recovered crashed replicas. In this paper, dynamic crash failures are assumed to be independent and identically distributed (i.i.d.) random variables for all processes during each phase.

Since our model is derived from the communication pattern of the protocol, special roles such as the primary in PBFT which follow a different communication pattern, are incorporated into the model.

B. Network Model

We assume that each network node has a peer-to-peer connection to all other nodes. The network model in this paper allows for (i) messages to be delayed indefinitely, i.e., past the configured timeout parameter of PBFT, and (ii) message omissions as well as corruptions, as they may appear in e.g. wireless networks. The former case acknowledges BFT protocols that rely on synchronous periods to guarantee liveness and are based on timeouts to detect process and/or link failures. PBFT, for example, makes use of timeouts to detect if progress is being made and as a consequence to initiate the view-change protocol. Messages that arrive after a configured timeout can therefore be considered as message omissions. The same applies to invalid or corrupted messages. The resulting failure model can be described with the *dynamic link failure model* [9]. While in practice many BFT protocols rely on the network layer to guarantee reliable communication, e.g., TCP, they should implement means to handle lost messages due to crashed or malicious processes. We therefore assume unidirectional links, which implies unreliable communication.

If assumptions made regarding the bound of message delays fail, i.e., the timeouts are not configured appropriately, the protocol can be considered to operate in an asynchronous network with unbounded message delays. This does not apply if an attacker is considered to have control over the scheduling of messages, as this could easily lead to stopping a BFT protocol altogether [27]. As with process failures, the link failures are assumed to be i.i.d. for all links.

IV. Modeling PBFT

The model presented in this section offers means to evaluate PBFT in the presence of dynamic link failures and crash failures. For the sake of clarity, we provide an overview of our modeling approach and introduce our notation first. Next, we unroll our model for various failure types step-by-step starting with dynamic crash failures, before we incorporate dynamic link failures.

A. Overview

In PBFT, the happy path consists of five phases of message exchanges, as depicted in Figure 1. The first and last phase consist of transmissions from and to the client. In the first phase, the leader of the current view will collect and serialize client requests. This is followed by a phase in which the primary will disseminate the requests to all other replicas in so-called `pre-prepare` messages. If a replica receives and accepts a `pre-prepare` message, it stores that message and enters the third phase, broadcasting and collecting a quorum, i.e., at least $2f + 1$, of `prepare` messages that match the stored `pre-prepare` message. The fourth phase mirrors the third phase, except with `commit` messages. If a quorum of valid `commit` messages is collected, the node will commit (and execute) the state transition. In the fifth and last phase of the protocol, replicas reply to the client, confirming that the client's request was executed from the replicated system.

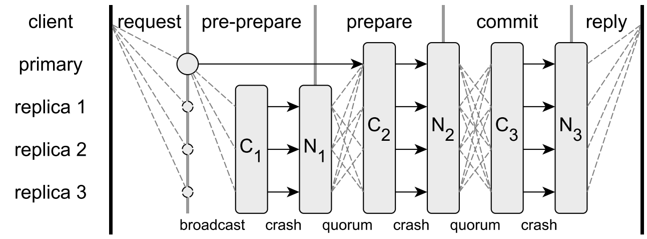


Fig. 1: Modeled view on PBFT's happy path communication pattern. Each phase is modeled via alternating predictions for crash (C_i) and link failures (N_i).

Omitting client interactions, the first and last phase can be disregarded and PBFT's happy path can be reduced to three phases. These phases can be summarized as a broadcast phase and two quorum collection phases. In the following, we assume that the primary is ready to initiate the consensus algorithm. Consequently, only the communication between the replicas is captured in our model.

In each phase of the protocol, the communication between and the availability of replicas is modeled as a combination of Bernoulli trials. More specifically, we model link and crash failures in alternating rounds for each of PBFT's phases, as depicted at the bottom of Figure 1. We use random variables N_i and C_i to express the success probabilities for the respective failure type in phase i.

In a first step, only faulty nodes are modeled as crash failures in a series of interdependent Bernoulli trials, i.e., $N_1 \rightarrow N_2 \rightarrow N_3$. In a second step, we extend the model by incorporating link failures. The communication is modeled along the lines of the three transmission phases C_1, C_2, and C_3. Combined with the node failures, our model yields an interleaving series of dependent system states, i.e., $C_1 \rightarrow N_1 \rightarrow C_2 \rightarrow N_2 \rightarrow C_3 \rightarrow N_3$.

In summary, the system state of all replicas at each protocol phase is captured by a series of probability density functions (PDFs), each constituting the calculation of the following. Please note, that each PDF allows for precise prediction of the protocol behavior and can be transformed into more common performance metrics, e.g., latency, with statistics or other models that predict the duration of individual phases.

B. Notation

In Table I, we summarize relevant probabilities, events, and random variables, which are used in our model. For ease of comprehension, the link and crash failure distributions are now reduced to single probabilities, i.e., p_l and p_c, respectively. The assumption to have identical link failure probabilities for all links is not an uncommon practice in this field of research [21], [23], [24]. The system state of the protocol is modeled by calculating PDFs that describe each replica's state. To this end, the random variables and events listed in Table I are indexed according to PBFT's phases. In particular, C_1, C_2, and C_3 represent the number of replicas that received a `pre-prepare` message, received a quorum of `prepare` messages and received a quorum of `commit` messages, respectively. Additionally, the number of active replicas after each phase is described with N_1, N_2, and N_3. Due to the nature of the PBFT algorithm, the distributions are dependent on each other, i.e., a replica that crashed or failed to collect the required messages will not be able to complete the happy path.

A key building block of our model are Bernoulli trials. Therefore, we use the notation $B(n,p,k) = \binom{n}{k} \cdot p^k \cdot (1-p)^{n-k}$ to express the probability to get exactly k successes in a Bernoulli experiment with n trials and a success probability of p. Furthermore, we define $B(n,p,[k,l]) = \sum_{i=k}^{l} B(n,p,i)$ as the sum over all Bernoulli trials with at least k and up to l successes. Finally, the notations $P_X(x) = P(X = x)$ and $P_{X|Y}(x|y) = P(X = x|Y = y)$ are used to abbreviate (conditional) probabilities.

C. Modeling crash failures

We start by modeling one of the most prominent failures, crash failures. In particular, a crash failure implies no participation of the crashed process in the phase in which the crash occurred. Hence, the replica will neither receive nor send any messages. In the following, the three random variables N_1, N_2, and N_3 are derived for a crash failure probability p_c, assuming reliable communication links.

The happy path of PBFT is initiated with the primary broadcasting a `pre-prepare` message to all other replicas. For the sake of simplicity, it is assumed the primary cannot crash during the `pre-prepare` phase. Since the probability for a crash is uniform across all nodes, the PDF of the still active replicas N_1 is given by $P_{N_1}(n_1) = B(n, 1-p_c, n_1)$. The distribution of N_1 describes the number of replicas that will now broadcast `prepare` messages to all other replicas in the second phase of the protocol.

TABLE I: Model notation for PBFT.

Symbol	Description	Range
	PROBABILITIES	
p_c	Probability for a crash failure.	$[0,1]$
p_l	Probability for a link failure.	$[0,1]$
	EVENTS	
C_p	Indicates a successful reception of a quorum of `prepare` messages at the current primary.	
	RANDOM VARIABLES	
C_1	The number of replicas that have received a `pre-prepare` message.	$[0, n-1]$
N_1	The number of replicas, excluding the primary, that did not crash in the `pre-prepare` phase.	$[0, n-1]$
$C_{2,n}$	The number of replicas, excluding the primary, that received a `pre-prepare` message as well as collected a quorum of `prepare` messages.	$[0, n-1]$
C_2	The number of replicas that received a `pre-prepare` message as well as collected a quorum of `prepare` messages.	$[0, n]$
N_2	The number of replicas that did not crash in the `prepare` phase.	$[0, n]$
C_3	The number of replicas that have received a `pre-prepare` message as well as collected a quorum of both, `prepare` and `commit` messages.	$[0, n]$
N_3	The number of replicas that have did not crash in the `commit` phase and successfully executed the algorithm.	$[0, n]$

Following this procedure, the distribution of active nodes in further phases is calculated conditioned on the previous phase, meaning

$$P_{N_i}(n_i) = \sum_{n_{i-1}} B(n_{i-1}, 1-p_c, n_i) \cdot P_{N_{i-1}}(n_{i-1}). \qquad (1)$$

for $i = 2, 3$. Adding up the values of $P(N_3 \geq 2f+1)$ allows to predict the success probability for the happy path of PBFT. As replicas cannot skip a phase in PBFT, a crashed replica will not recover during the happy path rendering dynamic crashes similar to permanent ones.

D. Modeling crash and link failures

We now extend our model by introducing link failures, i.e., the links are no longer considered reliable and are subject to a link failure probability p_l. The three random variables C_1, C_2, and C_3 are introduced to model the behavior of the protocol during the three communication phases. Due to the special behavior of the primary in the second phase, C_2 is divided into the event C_p and random variable $C_{2,n}$ to capture the communication of the primary and other replicas, respectively. Assuming the dynamic link failure model, all links are subject to the same failure probability p_l and can be, as with crash failures before, described with Bernoulli trials. In the following, we therefore start alternating between C_i and N_i (cf. Figure 1) to model the success of the message delivery and node availability, respectively.

Calculating C_1: The primary broadcasts a pre-prepare message to all other replicas. Since the success probability for each message transmission is equal to $1-p_l$ and independent of other transmissions, the number of successful transmissions can be calculated with a Bernoulli trial. The PDF of C_1 is given by $P_{C_1}(c_1) = B(n-1, 1-p_l, c_1)$ and describes the number of replicas that have received a pre-preapre message from the primary.

Calculating N_1: Based on the distribution of C_1, some replicas might crash in this phase, leading to

$$\begin{aligned} P_{N_1}(n_1) &= \sum_{c_1=0}^{n-1} P_{N_1|C_1}(n_1 \,|\, c_1) \cdot P_{C_1}(c_1) \\ &= \sum_{c_1=0}^{n-1} B(c_1, 1-p_c, n_1) \cdot P_{C_1}(c_1). \end{aligned} \tag{2}$$

Calculating C_2: The communication in the second phase is composed of the following: (i) whether the primary can collect $2f$ prepare messages (i.e., event C_p) (ii) the number of non-primary replicas that collect at least $2f+1$ prepare messages (i.e., $C_{2,n}$).

The primary can only collect at least $2f$ prepare messages if at least $2f$ active replicas have received the previous pre-prepare message, i.e., $N_1 \geq 2f$. In this case, at least $2f$ transmissions of prepare messages of the N_1 replicas have to successfully reach the primary. This can be expressed as the sum over all favorable Bernoulli trials, i.e., all trials with at least $2f$ successes out of N_1. The conditional probability $P(C_p \,|\, N_1 = n_1)$ for the primary to collect the prepare message is given by

$$P(C_p \,|\, N_1 = n_1) = \begin{cases} 0, & n_1 < 2f \\ B(n_1, 1-p_l, [2f, n]) & \text{otherwise.} \end{cases} \tag{3}$$

For a non-primary node, i.e., a replica, to advance to C_2, two requirements need to be met: (i) the replica has received a respective pre-prepare message, and (ii) the replica has collected a quorum of matching prepare messages. For a quorum, only $2f-1$ prepare messages are required, since a replica's own prepare message and the primary's pre-prepare message count towards the $2f+1$ required messages. The previous requirements translate to

1) there cannot be more replicas that receive $2f-1$ prepare messages than replicas that have previously received a pre-prepare message, i.e., $C_{2,n} \leq N_1$, and
2) a replica can only receive $2f-1$ prepare messages if at least $2f$ replicas, including itself, have received a pre-prepare, i.e., $N_1 \geq 2f$.

The calculation of $C_{2,n}$ can thus be divided into the following cases, assuming that n_1 replicas have received a pre-prepare message. First, for $n_1 < 2f$, no replica will be able to gather the required quorum of prepare messages, thus, the probability for $c_{2,n} = 0$ is always one. Second, if $c_{2,n} > n_1$, the probability has to be zero. Finally, for all other cases, of the n_1 replicas that broadcast prepare messages, excluding the primary, the probability for $c_{2,n}$ replicas to receive $2f-1$ of those messages can be modeled as another Bernoulli trial. The probability of success in that Bernoulli trial is identical to a replica receiving at least $2f-1$ messages of the n_1-1 possible. Thus, the conditional PDF of $C_{2,n}$ for $n_1 \geq 2f$ and $c_{2,n} \leq n_1$ is given by

$$P_{C_{2,n}|N_1}(c_{2,n} \,|\, n_1) = B(n_1, p_2(n_1), c_{2,n}) \tag{4}$$

with $p_2(n_1)$ being the probability that a replica will receive at least $2f-1$ prepare messages, given that n_1 replicas, including the replica itself, are broadcasting that message, which implies they have received the pre-prepare message as well. This can be calculated with another Bernoulli trial to get $2f-1$ receptions from n_1-1 messages of the other replicas: $p_2(n_1) = B(n_1-1, 1-p_l, [2f-1, n])$.

Combining (3) and (4) yields the conditional PDF of C_2. The calculation is split into multiple cases as follows

$$P_{C_2|N_1}(c_2|n_1) = \begin{cases} P_{C_{2,n}|N_1}(0 \,|\, n_1) \cdot P(\overline{C_p} \,|\, N_1 = n_1), & c_{2,n} = 0 \\ P_{C_{2,n}|N_1}(n-1 \,|\, n_1) \cdot P(C_p \,|\, N_1 = n_1), & c_{2,n} = n \\ P_{C_{2,n}|N_1}(c_{2,n} \,|\, n_1) \cdot P(\overline{C_p} \,|\, N_1 = n_1) & \\ + P_{C_{2,n}|N_1}(c_{2,n}-1 \,|\, n_1) \cdot P(C_p \,|\, N_1 = n_1), & c_{2,n} \leq n_1 + 1 \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

The final PDF of C_2 is given by applying the law of total probability to (5), which yields $P_{C_2}(c_2) = \sum_{n_1=0}^{n-1} P_{C_2|N_1}(c_2 \,|\, n_1) \cdot P_{N_1}(n_1)$.

Calculating N_2: As with N_1 and (2), the distribution of replicas that are still active, based on C_2, is $P_{N_2}(n_2) = \sum_{c_2=0}^{n} B(c_2, 1-p_c, n_2) \cdot P_{C_2}(c_2)$.

Calculating C_3: Now, let us turn to the states C_3 and N_3. In the third phase, the primary behaves in the same way as every other replica, simplifying many calculations regarding the communication as we do not need to mind so many exceptions. As with C_2, there are two requirements necessary for a replica to reach C_3: (i) the replica must be in state C_2 and (ii) it must have received at least $2f$ commit messages, not counting its own. Thus, we can conclude that

1) there cannot be more replicas that have received $2f$ commit messages than replicas that have reached C_2, i.e., $C_3 \leq C_2$, and
2) a replica can only receive $2f$ commit messages if at least $2f+1$ replicas, including itself, have reached state C_2, i.e., $C_2 > 2f$.

Deriving C_3 is similar to C_2. The conditional probability of C_3 for $c_2 > 2f$ and $c_3 \leq c_2$ is accordingly

$$P_{C_3|C_2}(c_3 \,|\, c_2) = B(c_2, p_3(c_2), c_3) \tag{6}$$

where $p_3(c_2)$ is the probability that a replica will receive at least $2f$ `commit` messages if c_2 replicas, including itself, are broadcasting that message, i.e.,

$$p_3(c_2) = B(c_2 - 1, 1 - p_l, [2f, n]). \tag{7}$$

Applying the law of total probability yields $P_{C_3}(c_3) = \sum_{c_2=0}^{n} P_{C_3|C_2}(c_3 \mid c_2) \cdot P_{C_2}(c_2)$.

Calculating N_3: Finally, $P_{N_3}(n_3) = \sum_{c_3=0}^{n} B(c_3, 1 - p_c, n_3) \cdot P_{C_3}(c_3)$, which denotes the PDF of all active nodes after the last phase.

If more than $2f$ replicas have completed the last phase, i.e., $P(N_3 > 2f)$, the happy path of PBFT was successful. For the system to provide liveness in regards to the current request, only $f + 1$ replicas are sufficient.

E. Generalization

For the sake of simplicity, we so far assumed constant failure probabilities for links and processes, i.e., p_l and p_c. We also assumed in our calculations, that those probabilities be constant for each phase of PBFT. This is not a requirement and could be expanded to reflect more sophisticated failure models that include time-based correlations as long as they remain i.i.d. for each phase.

Since the model is derived solely from communication patterns, it can be adapted to other fault tolerant protocols. This is facilitated by the modular design of the model, i.e., the expression of communication phases, e.g., broadcast, quorum, as PDFs which can be combined to describe the overall system state. To demonstrate the adaptability, we show the application of the model to BFT-SMaRt, Zyzzyva and SBFT. The detailed adaptations are available in the extended pre-print of this paper [28], where we showcase how the model can be applied to a variety of communication patterns, including client interaction and the possibility to branch into a fast or slow path.

We deliberately chose to highlight the model derivation in this section, leaving the formal definition of the modular components for future work.

V. Model Validation

To verify the correctness of the model, a discrete-event simulator was written in Rust. The simulation is publicly available on Github[1]. In a first instance, the simulator implements the happy path of PBFT for single requests without batching. The dynamic link failure model is realized by discarding each message reception event with a configurable probability p_l. In addition, each node will miss all messages belonging to a certain communication phase with probability p_c, simulating a crash failure. By doing this, we can compare the simulated state with the predictions of our model. The simulation can easily scale to larger numbers of nodes (above 100) since only the state transitions in the happy path are of interest and no actual SMR is implemented, i.e., requests are not executed.

In order to validate our model with an independent source, we also deployed the Java-based public BFT SMR library BFT-SMaRt [4] as a reference implementation. It implements a consensus protocol that bears a high resemblance to PBFT: it utilizes epochs, an equivalent to the views in PBFT, and operates in three phases with respective message types [29]. For simplicity, we stick to PBFT's terminology, when discussing BFT-SMaRt While mostly similar, the communication pattern of BFT-SMaRt differs from PBFT in two details, which required minor model adaptations. Firstly, nodes do not count the primary's `pre-prepare` message as a `prepare` message for the second phase. Secondly, nodes are allowed to skip the second phase if a quorum of other nodes were able to complete that phase. We describe the model adaptations in the extended pre-print of this paper [28].

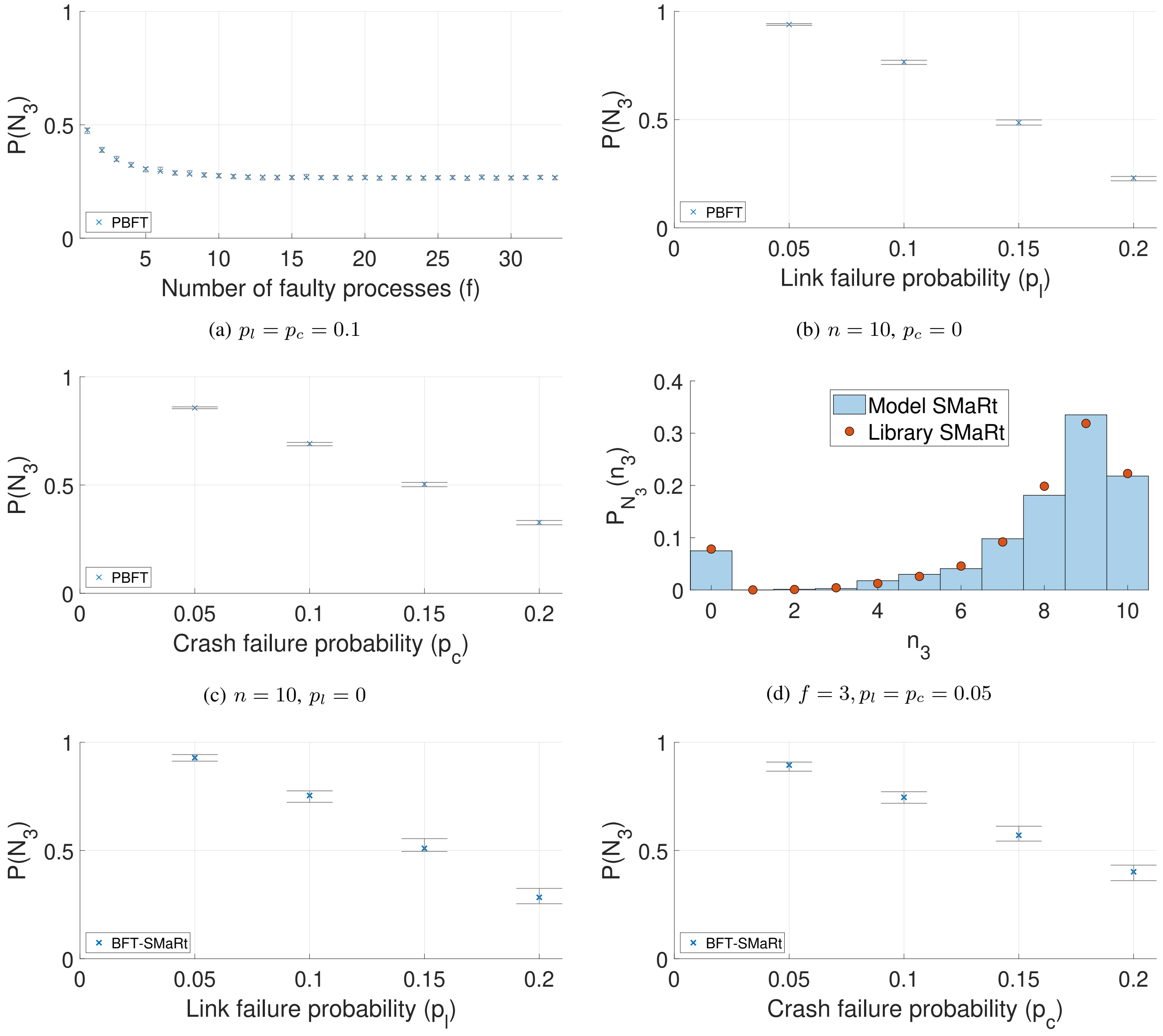
In order to apply dynamic link and crash failures to BFT-SMaRt, artificial message omission probabilities were implemented into the library. Accordingly, all messages are dropped with the probability p_l and each node discards all messages of a whole phase with probability p_c. The changes necessary to implement the aforementioned failures affected the class that is responsible for handling incoming messages only and consisted of less than 50 lines of additional code. The library was executed on a single computer and up to 10 replicas and one client were instantiated to execute the requests.

Increasing the number of processes while keeping the maximum number of faulty processes constant leads to an increased robustness of both protocols against link and crash failures, because more messages are available to build a quorum while the required quorum size remains equal. We therefore evaluate both protocols for the most interesting scenario $n = 3f + 1$, i.e., the minimum number of processes required to tolerate f faulty processes.

To validate the model for a larger parameter space, we evaluated PBFT for different numbers of processes, link failure rates, and crash failure rates. Figures 2a to 2c show the probability of a single (representative) process to successfully reach phase N_3 for different n, p_l and p_c, respectively. The simulation results for 5,000 protocol executions are plotted with 99% confidence intervals and model predictions are depicted as crosses. Increasing the number of processes in the network can have, depending on the number of processes and failure rates, either a stabilizing or destabilizing effect on the performance. A more detailed analysis of this behavior is given in Section VI-C. Increasing the failure rate of either links or processes causes a constant decrease for $P(N_3)$.

The comparison between model predictions and experimental results for BFT-SMaRt are shown in Figures 2d to 2f. Since BFT-SMaRt implements actual SMR and the execution was unstable due to the previously mentioned halts during the view-change protocol, we evaluated 1,000 protocol executions for each parameter combination only (without batching). Figure 2d depicts the measured and by the model predicted PDF of $P(N_3)$. The impact of link and crash failures on BFT-SMaRt is similar to PBFT. The small deviations visible between Figures 2b and 2c and Figures 2e and 2f stem from the

[1] https://github.com/mani2416/bft_simulation



(a) $p_l = p_c = 0.1$

(b) $n = 10$, $p_c = 0$

(c) $n = 10$, $p_l = 0$

(d) $f = 3, p_l = p_c = 0.05$

(e) $n = 10$, $p_c = 0$

(f) $n = 10$, $p_l = 0$

Fig. 2: Model validation results for PBFT (a-c) and BFT-SMaRt (d-f).

algorithmic differences described above. The overall results confirm that our model predictions for PBFT and BFT-SMaRt align accurately with the simulations and experimental results.

VI. Evaluation

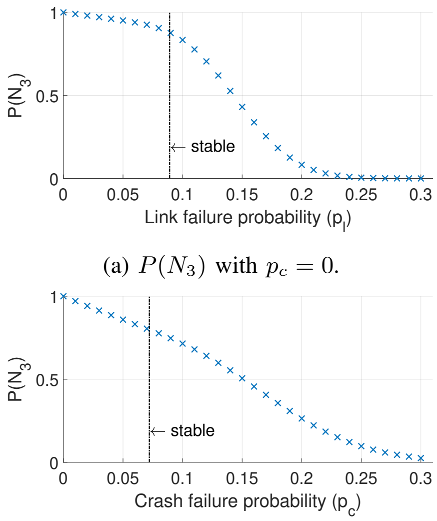
A. Protocol stability

The quorum collection phase in fault tolerant protocols ,i.e., for $2f+1$ processes to collect $2f$ out of $3f$ possible messages (not counting its own), is inherently resilient against link failures. A node cannot collect a quorum if at least $f+1$ out of its $3f$ incoming links are failing. Consequently, even in the worst case, at least $f+1$ nodes with at least $f+1$ link failures, i.e., $(f+1)^2$ overall link failures, are necessary for the quorum collection phase to potentially fail.

Given our model assumptions, we can calculate the theoretical failure rate necessary for a quorum phase in PBFT to fail. Since the number of processes that partake in each quorum phase is dependent on previous phases, the boundary for each phase is calculated as

$$\frac{((f+1)-(n-E[N_{i-1}]))^2}{E[N_{i-1}](E[N_{i-1}]-1)} \tag{8}$$

with $E[N_{i-1}]$ being the expected number of nodes that are still currently active. Depicted in Figure 3 are the predicted probabilities for $P(N_3)$ for increasing link failures (Figure 3a) and crash failures (Figure 3b). The line labeled "stable" marks



(a) $P(N_3)$ with $p_c = 0$.

(b) $P(N_3)$ with $p_l = 0$.

Fig. 3: $P(N_3)$ for PBFT with $n = 25$.

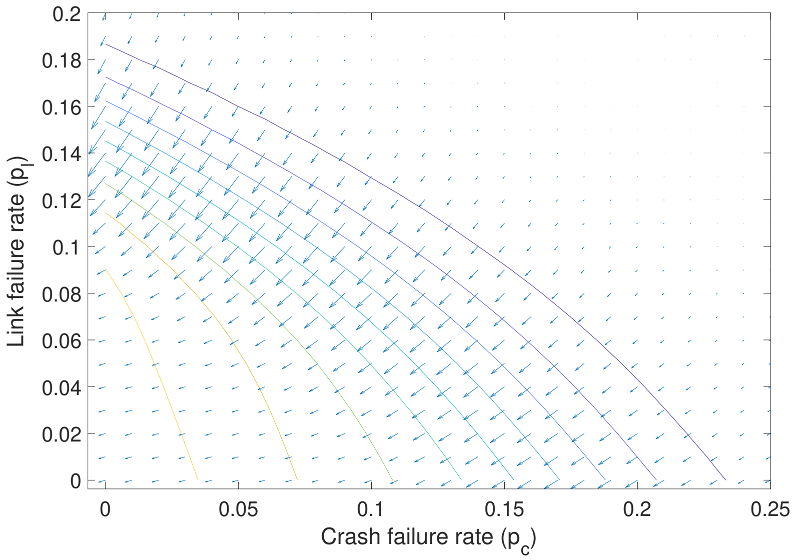


Fig. 4: Contour plot of $P(N_3)$ as predicted by our model for PBFT with $n = 40$; the gradient shows a vector field over p_c and p_l.

the boundary given by (8). The linear decrease to the left of the boundary in Figure 3a originates from the previous phases of the protocol. Since the first phase implements a one-to-all broadcast, the failing nodes will increase linearly with the failure rates. The same effect is even more pronounced for increasing crash failures in Figure 3b, albeit with an even steeper linear phase. Because processes cannot recover within the happy path of PBFT, each successive phase with crash failures will decrease the number of available nodes for further phases, leading to the steeper decline before the boundary.

The evaluation methodology and the respective results can be used to parameterize the protocol to ensure that the protocol execution remains stable even for a given failure rate. Since most BFT protocols treat delayed messages as link failures, the model can, e.g., be utilized to fine-tune timeouts. That is, for a given delay distribution, a timeout parameter can be translated to a failure rate. A small timeout leads accordingly to a higher failure rate, but at the same time is able to quickly detect (genuinely) lost messages and make progress. For instance, let us assume that the message delay on all links can be described with a normal distribution of mean $\mu = 100\,\text{ms}$ and standard deviation $\sigma = 10\,\text{ms}$. Further, we assume the result of (8) to be 0.1 for some arbitrary protocol. The timeout that keeps the protocol in the stable region is derived by finding an upper bound, where the integrated PDF of the delays is equal to $1 - b_{\text{stable}} = 0.9$. In our example, the timeout should be $> 87.19\,\text{ms}$. To conclude, our model allows to evaluate various failure scenarios and adjust parameters accordingly.

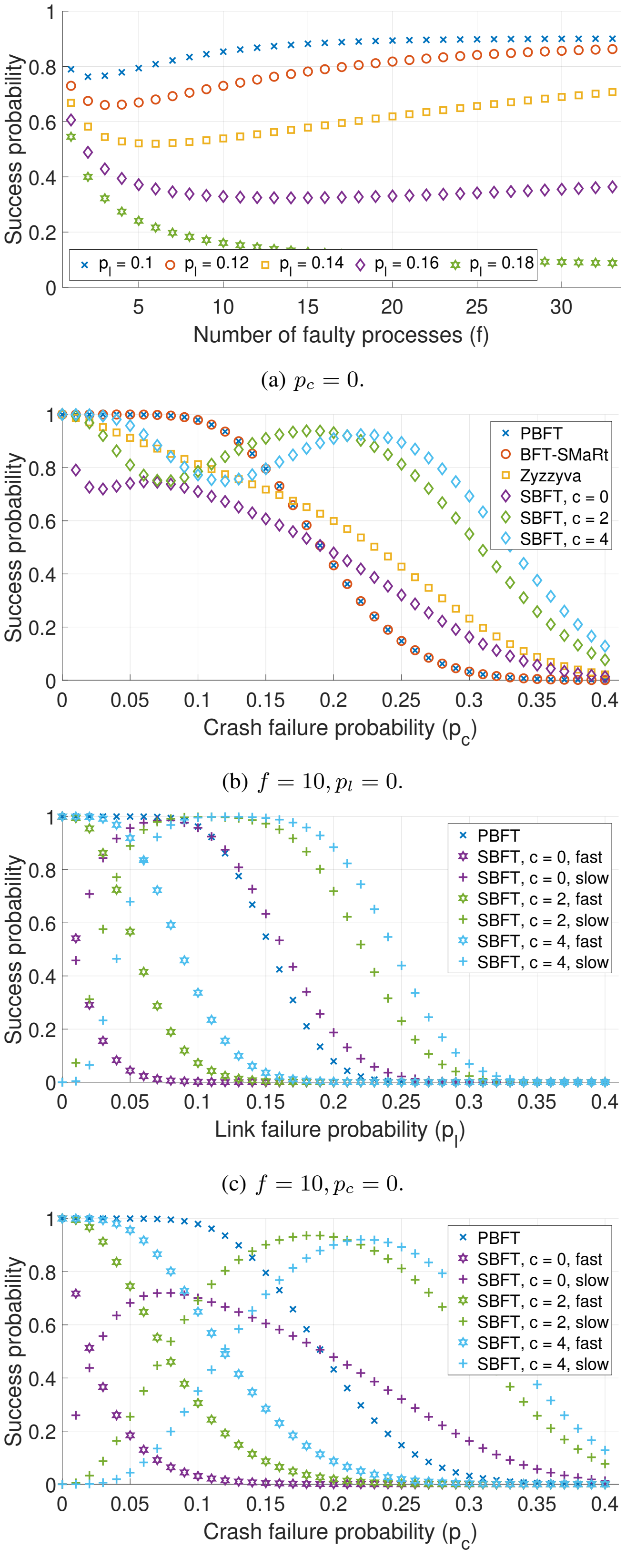
B. Impact of number of processes, link and crash failures

To better demonstrate the predictive capabilities of the model, a contour plot of $P(N_3)$ for PBFT is provided in Figure 4, for varying link and crash failure rates. Additionally, the gradient is displayed, derived from the operating points for different p_l and p_c, as they are predicted by the model. The orientation of the arrows indicates the impact of variations in either failure rate on $P(N_3)$. The more pronounced the horizontal component of a vector, the more dominant is the impact of crash failures on $P(N_3)$ and the same applies to the vertical component and link failures. The contour plot allows to quickly discern the impact of either failure rate on the protocol. Figure 4 shows that for low link failure rates, changes in the crash failure rate will dominate the success probability of the protocol, while for very low rates of crash failures and a moderate number of link failures ($p_l > 0.1$), the link failure rate dominates. Figure 4 also validates the observations made in Figure 3, i.e., the crash failure rates dominates the linear decline before the stable lines, while the link failures gain in impact for higher failure rates.

Although it is well known that most BFT protocols do not scale well with the number of processes due to the quadratic message complexity, it generally offers means to increase stability in the presence of dynamic link failures. In the extended version of the paper [28], we show that the probability to collect a quorum for $n \to \infty$ converges to 0 or 1, depending on p_l and the quorum size. As a consequence, the number of nodes can increase the success probability of the quorum collection phase for failure rates below a certain threshold.

C. Comparison: PBFT, BFT-SMaRt, Zyzzyva, and SBFT

To showcase the adaptability of our model, we applied it to Zyzzyva [10] and SBFT [11]. An exemplary comparison of all protocols for different crash failure rates is given in Figure 5b. Depicted are the overall success probabilities, i.e., for Zyzzyva and SBFT the combination of fast and slow



(a) $p_c = 0$.

(b) $f = 10, p_l = 0$.

(c) $f = 10, p_c = 0$.

(d) $f = 10, p_l = 0$.

Fig. 5: (a) happy path success probabilities of PBFT, BFT-SMaRt, Zyzzyva and SBFT dependent on crash failure rates, (c) and (d): detailed analysis of SBFT.

paths. To better demonstrate the capabilities of the model, the individual success probabilities for the fast and slow path of SBFT for different link and crash failure rates are plotted in Figures 5c and 5d. Since SBFT allows for optional, additional replicas, denoted as c, the model allows to quickly assess the protocol behavior for different failure rates and configurations of c. The plots show that SBFT outperforms PBFT for increasing numbers of c and higher failure rates, while PBFT is more stable if SBFT transitions from the fast path to the slow path.

VII. Conclusion

The probabilistic predictions of the presented model were validated with implementations of PBFT and BFT-SMaRt for various numbers of processes and dynamic link and crash failure rates. It was demonstrated with BFT-SMaRt, Zyzzyva and SBFT, that the model can be adapted with little effort to other communication patterns of BFT protocols. The model gives a prediction of the distribution of process states during execution, allowing prediction of protocol behavior (e.g. how many view changes will occur) and therefore performance evaluation. Additionally, if the message delay statistics are known, the model can be deployed to tune the timeouts for BFT protocols, since most protocols cannot differentiate between a delayed or an omitted message, making them indifferent in their impact on the algorithm. The model allows to assess the impact of crash and link failures for various operating points of a protocol to identify key boundaries regarding protocol stability.

As was demonstrated with BFT-SMaRt, Zyzzyva and SBFT, the model can be applied to different BFT protocols by modifying the respective equations for the distributions or adding further random variables should the protocol consist of more phases (as is the case with SBFT). Further adaptations are facilitated by the fact that a body of BFT protocols are derived from the core structure of PBFT and consist of interdependent phases.

In further work, we are planning to apply the model to more BFT protocols and evaluate their performance regarding dynamic failures. Furthermore we are exploring possibilities to extend the model to predict more sophisticated key performance indicators, such as throughput and latency. Lastly, we will consider adaptations to our model in order to account for correlated link failures, e.g., as was proposed with a model by Nguyen [30].

References

[1] F. Thiel, M. Esche, F. Grasso Toro, A. Oppermann, J. Wetzlich, and D. Peters, "The European Metrology Cloud," in *Proceedings of the 18th International Congress of Metrology*, Jan. 2017. doi: 10.1051/metrology/201709001

[2] M. Castro and B. Liskov, "Practical Byzantine Fault Tolerance and Proactive Recovery," *ACM Trans. Comput. Syst.*, vol. 20, no. 4, pp. 398–461, Nov. 2002. doi: 10.1145/571637.571640

[3] P. Sazonova, "The general universal model of blockchain technology based on an analysis of some implementations," in *Communication Papers of the 2020 Federated Conference on Computer Science and Information Systems*. PTI, Sep. 2020. doi: 10.15439/2020f190. [Online]. Available: https://doi.org/10.15439/2020f190

[4] A. Bessani, J. Sousa, and E. E. P. Alchieri, "State Machine Replication for the Masses with BFT-SMART," in *2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, Jun. 2014. doi: 10.1109/DSN.2014.43. ISSN 1530-0889 pp. 355–362.

[5] E. Buchman, J. Kwon, and Z. Milosevic, "The latest gossip on BFT consensus," *CoRR*, vol. abs/1807.04938, 2018.

[6] P. Aublin, S. B. Mokhtar, and V. Quéma, "RBFT: Redundant Byzantine Fault Tolerance," in *2013 IEEE 33rd International Conference on Distributed Computing Systems*, Jul. 2013. doi: 10.1109/ICDCS.2013.53. ISSN 1063-6927 pp. 297–306.

[7] R. Kapitza, J. Behl, C. Cachin, T. Distler, S. Kuhnle, S. V. Mohammadi, W. Schröder-Preikschat, and K. Stengel, "CheapBFT: Resource-efficient Byzantine Fault Tolerance," in *Proceedings of the 7th ACM European Conference on Computer Systems*, ser. EuroSys '12. New York, NY, USA: ACM, 2012. doi: 10.1145/2168836.2168866. ISBN 978-1-4503-1223-3 pp. 295–308.

[8] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolić, S. W. Cocco, and J. Yellick, "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains," in *Proceedings of the Thirteenth EuroSys Conference*, ser. EuroSys '18. New York, NY, USA: ACM, 2018. doi: 10.1145/3190508.3190538. ISBN 978-1-4503-5584-1 pp. 30:1–30:15.

[9] N. Santoro and P. Widmayer, "Time is not a healer," in *STACS 89*, B. Monien and R. Cori, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1989. doi: 10.1007/BFb0028994. ISBN 978-3-540-46098-5 pp. 304–313.

[10] R. Kotla, L. Alvisi, M. Dahlin, A. Clement, and E. Wong, "Zyzzyva: Speculative Byzantine Fault Tolerance," *SIGOPS Oper. Syst. Rev.*, vol. 41, no. 6, pp. 45–58, Oct. 2007. doi: 10.1145/1323293.1294267

[11] G. Golan Gueta, I. Abraham, S. Grossman, D. Malkhi, B. Pinkas, M. Reiter, D. Seredinschi, O. Tamir, and A. Tomescu, "Sbft: A scalable and decentralized trust infrastructure," in *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2019. doi: 10.1109/DSN.2019.00063 pp. 568–580.

[12] G. Bracha and S. Toueg, "Asynchronous Consensus and Broadcast Protocols," *J. ACM*, vol. 32, no. 4, pp. 824–840, Oct. 1985. doi: 10.1145/4221.214134

[13] C. Dwork, N. Lynch, and L. Stockmeyer, "Consensus in the Presence of Partial Synchrony," *J. ACM*, vol. 35, no. 2, pp. 288–323, Apr. 1988. doi: 10.1145/42282.42283

[14] M. J. Fischer, N. A. Lynch, and M. S. Paterson, "Impossibility of Distributed Consensus with One Faulty Process," *J. ACM*, vol. 32, no. 2, pp. 374–382, Apr. 1985. doi: 10.1145/3149.214121

[15] B. Liskov, "From Viewstamped Replication to Byzantine Fault Tolerance," in *Replication: Theory and Practice*, ser. Lecture Notes in Computer Science, no. 5959, 2010.

[16] A. S. de Sá, A. E. Silva Freitas, and R. J. de Araújo Macêdo, "Adaptive Request Batching for Byzantine Replication," *SIGOPS Oper. Syst. Rev.*, vol. 47, no. 1, pp. 35–42, Jan. 2013. doi: 10.1145/2433140.2433149

[17] U. Schmid, "How to model link failures: a perception-based fault model," in *2001 International Conference on Dependable Systems and Networks*, Jul. 2001. doi: 10.1109/DSN.2001.941391 pp. 57–66.

[18] U. Schmid, B. Weiss, and I. Keidar, "Impossibility Results and Lower Bounds for Consensus under Link Failures," *SIAM Journal on Computing*, vol. 38, no. 5, pp. 1912–1951, 2009. doi: 10.1137/S009753970443999X

[19] R. Halalai, T. A. Henzinger, and V. Singh, "Quantitative Evaluation of BFT Protocols," in *2011 Eighth International Conference on Quantitative Evaluation of SysTems*, Sep. 2011. doi: 10.1109/QEST.2011.40 pp. 255–264.

[20] H. Sukhwani, J. M. Martínez, X. Chang, K. S. Trivedi, and A. Rindos, "Performance Modeling of PBFT Consensus Process for Permissioned Blockchain Network (Hyperledger Fabric)," in *2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS)*, Sep. 2017. doi: 10.1109/SRDS.2017.36 pp. 253–255.

[21] A. Singh, T. Das, P. Maniatis, P. Druschel, and T. Roscoe, "BFT Protocols Under Fire," in *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, ser. NSDI'08. Berkeley, CA, USA: USENIX Association, 2008. ISBN 111-999-5555-22-1 pp. 189–204.

[22] M. Abd-El-Malek, G. R. Ganger, G. R. Goodson, M. K. Reiter, and J. J. Wylie, "Fault-scalable Byzantine Fault-tolerant Services," *SIGOPS Oper. Syst. Rev.*, vol. 39, no. 5, pp. 59–74, Oct. 2005. doi: 10.1145/1095809.1095817

[23] N. Fathollahnejad, E. Villani, R. Pathan, R. Barbosa, and J. Karlsson, "On reliability analysis of leader election protocols for virtual traffic lights," in *2013 43rd Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop (DSN-W)*, Jun. 2013. doi: 10.1109/DSNW.2013.6615529. ISSN 2325-6664 pp. 1–12.

[24] W. Xu and R. Kapitza, "RATCHETA: Memory-Bounded Hybrid Byzantine Consensus for Cooperative Embedded Systems," in *2018 IEEE 37th Symposium on Reliable Distributed Systems (SRDS)*, Oct. 2018. doi: 10.1109/SRDS.2018.00021. ISSN 2575-8462 pp. 103–112.

[25] U. Schmid, B. Weiss, and J. Rushby, "Formally verified Byzantine agreement in presence of link faults," in *Proceedings 22nd International Conference on Distributed Computing Systems*, Jul. 2002. doi: 10.1109/ICDCS.2002.1022311. ISSN 1063-6927 pp. 608–616.

[26] M. Biely, U. Schmid, and B. Weiss, "Synchronous consensus under hybrid process and link failures," *Theoretical Computer Science*, vol. 412, no. 40, pp. 5602–5630, 2011. doi: 10.1016/j.tcs.2010.09.032 Stabilization, Safety and Security.

[27] A. Miller, Y. Xia, K. Croman, E. Shi, and D. Song, "The Honey Badger of BFT Protocols," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: ACM, 2016. doi: 10.1145/2976749.2978399. ISBN 978-1-4503-4139-4 pp. 31–42.

[28] M. Nischwitz, M. Esche, and F. Tschorsch, "Bernoulli meets pbft: Modeling bft protocols in the presence of dynamic failures," 2020.

[29] C. Cachin, "Yet another visit to Paxos," Apr. 2011.

[30] H. H. Nguyen, K. Palani, and D. M. Nicol, "Extensions of Network Reliability Analysis," in *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Jun. 2019. doi: 10.1109/DSN.2019.00023 pp. 88–99.

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 301–311

DOI: 10.15439/2021F89
ISSN 2300-5963 ACSIS, Vol. 25

A random forest-based approach for survival curves comparison: principles, computational aspects, and asymptotic time complexity analysis

Lubomír Štěpánek
Department of Statistics and Probability
Faculty of Informatics and Statistics
University of Economics
nám. W. Churchilla 4, 130 67 Prague, Czech Republic
lubomir.stepanek@vse.cz
&
Institute of Biophysics and Informatics
First Faculty of Medicine
Charles University
Salmovská 1, Prague, Czech Republic
lubomir.stepanek@lf1.cuni.cz

Filip Habarta
Department of Statistics and Probability
Faculty of Informatics and Statistics
University of Economics
nám. W. Churchilla 4, 130 67 Prague, Czech Republic
filip.habarta@vse.cz

Ivana Malá
Department of Statistics and Probability
Faculty of Informatics and Statistics
University of Economics
nám. W. Churchilla 4, 130 67 Prague, Czech Republic
malai@vse.cz

Luboš Marek
Department of Statistics and Probability
Faculty of Informatics and Statistics
University of Economics
nám. W. Churchilla 4, 130 67 Prague, Czech Republic
marek@vse.cz

***Abstract*—The log-rank test and Cox's proportional hazard model can be used to compare survival curves but are limited by strict statistical assumptions. In this study, we introduce a novel, assumption-free method based on a random forest algorithm able to compare two or more survival curves. A proportion of the random forest's trees with sufficient complexity is close to the test's p-value estimate. The pruning of trees in the model modifies trees' complexity and, thus, both the method's robustness and statistical power. The discussed results are confirmed using a simulation study, varying the survival curves and the tree pruning level.**

I. Introduction

COMPARING two or more survival curves is relatively common in many applied areas such as biomedicine, econometrics, management, and others. When the curves are statistically significantly different, it may help treat the groups that are the curves built by in appropriate (separate) ways.

As typical for survival analysis, the variable of our interest usually describes a (time) development of proportions of individuals who have not experienced the event of interest yet (until each considered time point) in each consecutive time point of the time period of our interest.

Such a time development is commonly plotted using orthogonal polygonal lines, also known as *survival curves* in a two-dimensional (survival) plot, sometimes called Kaplan-Meier plot [1]. Since groups of individuals that are about to be compared have their own time developments of non-experiencing the event of interest, one Kaplan-Meier survival plot may include more than only one curve, as shown in Fig. 1.

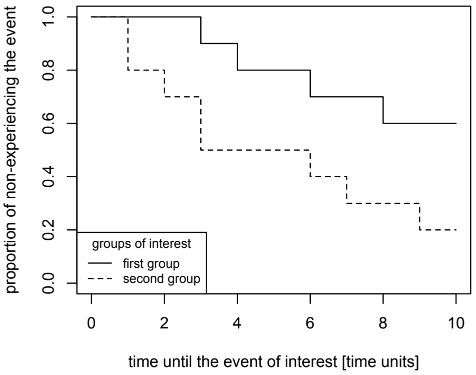


Fig. 1. Two time-to-event survival curves for two groups of interest in Kaplan-Meier (survival) plot.

Typically, following the logic behind the time development of non-experiencing the event of our interest, there are time points placed on the horizontal axis of a survival plot and proportions of subjects with no experience of the event of interest on a vertical axis.

Regardless of the total length of the time period of interest, it has in principle be finite and, consequently, we cannot get any piece of information whether the individuals not experiencing the event of interest in the time period would register the event after an end of the period, or not. That is why the time-to-event (survival) data are also called *right censored* data.

Since the experiencing of the event of interest is usually irreversible in time within the scope of classical survival analysis (the event is, e. g., a death, diagnosis, bankruptcy, failure, etc.), a subject registering the event whenever in the time period of the interest, continues to stay in this state till the end of the referenced time (the time of the right censoring). Thus, the survival curves are monotonous and nonincreasing. Intuitively, when the development of the event of interest differs between two groups, we may expect their survival curves are hardly similar and relatively far from each other within each considered time point.

If there are two survival curves to be statistically compared, the log-rank test as a tool of choice is usually performed [2] and commonly implemented, e. g. in R language and statistical environment [3] and its library `survival` [4]. However, a usage of the log-rank test is limited by statistical assumptions, that (i) censoring should not affect anyhow the observed events and (ii) the censoring should occur equally or at least near-equally in both compared groups, generating the survival curves. Also, (iii) the group's sizes are expected to be large enough, enabling the log-rank test's χ^2 statistics to fulfill its asymptotic properties.

To overcome the limitations done by the statistical assumptions of the original log-rank test or to increase its robustness or statistical power, several modifications of the traditional log-rank test were published. The first approach is to modify the hazard functions slightly, i. e., functions of rates of events based on fixed proportions of the events in the past, and relax their assumptions to increase their robustness as suggested by [5]. Then, another option is to introduce new covariates (variables) to enrich the model comparing two survival curves and increase the robustness, published in [6]. Also, employing various weighting schemes for individual observations, usually growing significance for earlier ones, may increase the statistical power of the test as investigated originally in [7] and then improved in [8], [9] and [10]. Finally, robust combinatorial and exact calculations of all possible combinations of the event experiencing and non-experiencing subjects for given total numbers of subjects in the compared groups are researched in [11] and [12] and using survival curves' finite combinatorial geometry in [13].

An advantage of the latter approach, based on robust combinatorial computations when comparing two survival curves, is that makes possible an estimation of asymptotic time complexity, as is in details commented by [14], [15], [16] and partly by [13], too.

When one wants to statistically compare three or more survival curves, there is an option to use Cox's proportional hazard model or a score-rank test based on Cox's model. Unfortunately, Cox's proportional model is also limited – it assumes that hazard proportions for each group are constant across all considered time points, which is often not met in practice. Some more robust versions of Cox's model were derived to minimize the violation of the constant hazards' ratio by real-world data, e. g. based on an idea of stratification of each group into subgroups according to their hazard similarity; however, those advanced models are usually limited by other, more complex assumptions [17], though.

The decision (or regression) trees and random forests are classical algorithms used for classification or regression problems. An idea to apply decision trees and random forest on survival tasks and right-censored, time-to-event data originate from [18], but initial thoughts rather aimed to a robust estimation of hazard functions' parameters, e. g. Nelson-Aalen estimator etc. The decision trees and random forests are naturally assumption-free robust, and fully non-parametric, especially in comparison to the log-rank test or the Cox's proportional hazard model, which is a property also utilized in this study and by the proposed alternative method for survival curves comparison.

The proceeding proceeds as follows. Firstly, in the section *Traditional methods for survival curves comparing and random forests revisited*, we shortly remind the fundamental principles of the log-rank test, Cox's proportional hazard model, decision trees, and random forests. We also discuss assumptions and limitations of the named methods that create room for new approaches that are less dependent on statistical assumptions.

Then, in the section *The proposed method for survival curves comparing*, we introduce a novel alternative for two or more survival curves comparing, based on random forest-based generating of multiple decision trees, using variables derived from original time-to-event data of compared groups of individuals in their nodes. The level of the trees' pruning is adjustable as a hyperparameter; it enables to control a complexity of the trees, i. e., an average number of nodes and leaves per tree in the forest. If a given tree in the random forest is able to classify whatever new observation in each of the groups (described by its survival curve), i. e., there exists at least one leaf node for each group assigning the observation to such a group, that tends to be contradictory the null hypothesis, claiming there are no statistical differences between the groups (and their survival curves). A proportion of the trees with sufficient complexity to all trees in the forest serves by definition as an estimate of p-value as would be analogously[1] returned by the traditional log-rank test, i. e., a conditional probability of collecting data as extreme or even more given there is no difference between the survival curves. Since the p-value is partially determined by the proportion of sufficiently complex trees to all trees of the random forest, the level of pruning may affect the robustness or statistical

[1] A numerical value of the p-value returned by the log-rank test and by the proposed method are not supposed to be equal, as discussed in the following sections.

power of the proposed random forest-based inference test, as is discussed more in details later.

The asymptotic time complexity of the p-value estimation, assuming the random forest model building, is then derived, and, finally, in the section *Simulation study*, a preliminary simulation study is performed to confirm the theoretically derived properties of the method. Besides others, the introduced approach offers a way how to compare more than two survival curves without any assumptions needed to be met.

II. Traditional methods for survival curves comparing and random forests revisited

Firstly, we remind principles and assumptions of the log-rank test and Cox's proportional hazard model that facilitates a better understanding of their limitations, which, consequently, opens room for improvements in survival curves comparing. We also recapitulate the logic of the decision trees and random forest heavily equipped in the proposed method for survival curves comparison.

A. Principles, assumptions, and limitations of the log-rank test

Principles of the log-rank test. Let's assume k distinct time points where the event of interest could take a place; the j-th time point is marked as t_j, where $j \in \{1, 2, 3, \ldots, k\}$, and all the time points are ordered in a tuple $(t_1, t_2, \ldots, t_k)^T$. Also, let's suppose there are two groups of subjects, marked by subscripts 1 and 2, respectively. For each of the time points, let's say for the j-th one (t_j) there are $r_{1,j}$ and $r_{2,j}$ individuals at risk (they have not experienced the event of interest yet or have been censored) in the group 1 and group 2, respectively, and $d_{1,j}$ and $d_{2,j}$ individuals who experienced the event in the group 1 and group 2, respectively. Thus, following the previous logic, we can construct a (contingency) table I.

TABLE I
NUMBERS OF INDIVIDUALS EXPERIENCING THE EVENTS OF INTEREST IN BOTH GROUPS (1 AND 2) AT TIME t_j.

	event of interest at the event time t_j		
group	yes	no	total
1	$d_{1,j}$	$r_{1,j} - d_{1,j}$	$r_{1,j}$
2	$d_{2,j}$	$r_{2,j} - d_{2,j}$	$r_{2,j}$
total	d_j	$r_j - d_j$	r_j

The log-rank test checks the null hypothesis H_0 that both groups experienced identical rates of the events of interest in time (also called *hazard functions*) [2], conditional on fixed rates in the past are the same. Under the null hypothesis H_0, the observed numbers of individuals experiencing the events could be considered as random variables $D_{1,j}$ and $D_{2,j}$ following a hypergeometric distribution with parameters $(r_j, r_{i,j}, d_j)$ for both $i \in \{1, 2\}$. Thus, the expected value of the variable $D_{i,j}$ is $\mathbb{E}(D_{i,j}) = r_{i,j}\frac{d_j}{r_j}$ and variance is $\text{var}(D_{i,j}) = \frac{r_{1,j} r_{2,j} d_j}{r_j^2}\left(\frac{r_j - d_j}{r_j - 1}\right)$ for both $i \in \{1, 2\}$. Finally, under the null hypothesis H_0, we can compare the observed numbers of events of interest, $d_(i,j)$, for all $j \in \{1, 2, 3, \ldots, k\}$, to their expected values $\mathbb{E}(D_{i,j}) = r_{i,j}\frac{d_j}{r_j}$. So, the test statistic for both $i \in \{1, 2\}$ is then

$$\chi^2_{\text{log-rank}} = \frac{\left(\sum_{j=1}^{k} d_{i,j} - \mathbb{E}(D_{i,j})\right)^2}{\sum_{j=1}^{k} \text{var}(D_{i,j})} = = \frac{\left(\sum_{j=1}^{k} d_{i,j} - r_{i,j}\frac{d_j}{r_j}\right)^2}{\sum_{j=1}^{k} \frac{r_{1,j} r_{2,j} d_j}{r_j^2}\left(\frac{r_j - d_j}{r_j - 1}\right)}, \tag{1}$$

which follows under H_0 a χ^2 distribution with 1 degree of freedom, $\chi^2_{\text{log-rank}} \sim \chi^2(1)$. For feasible large r_j, i. e. at least $r_j \geq 30$, a square root of $\chi^2_{\text{log-rank}}$ follows a standard normal distribution, $\sqrt{\chi^2_{\text{log-rank}}} \sim \mathcal{N}(0, 1^2)$. Since $\chi^2_{\text{log-rank}} \sim \chi^2(1)$, the statistics $\chi^2_{\text{log-rank}}$ can be uniquely transformed into p-value, which stands for a conditional probability of obtaining the test statistics $\chi^2_{\text{log-rank}}$ at least as extreme as the statistics actually observed, under the assumption that the null hypothesis H_0 reflects the reality.

Assumptions and limitations of the log-rank test. The right censoring of the data should not affect the occurrences of the event of interest in both groups anyhow. Also, the proportions of censored observations are supposed to be of (nearly) equal size in both groups. Otherwise, the test statistic $\chi^2_{\text{log-rank}}$ calculated using (1) could be biased for $i = 1$, or for $i = 2$.

Then, putting together the equation (1), so the test statistic $\chi^2_{\text{log-rank}}$ follows a χ^2 distribution, and the table I, both the initial total number of individuals r_0 at risk and initial number $r_0 - d_0$ not experiencing the event, should be large enough. Otherwise, so-called Cochrane criteria for minimal sample size for χ^2 tests are not met and the $\chi^2_{\text{log-rank}}$ statistics could not fulfill the χ^2 asymptotic properties; or, analogously, both the numerator and denominator of the statistics (1) are relatively small and an estimate of the $\chi^2_{\text{log-rank}}$ statistics is numerically unstable.

All the named issues may decrease the robustness or statistical power of the log-rank test.

Furthermore, by investigating the denominator of the equation (1), we can easily realize the test statistic $\chi^2_{\text{log-rank}}$ is the highest when the denominator $\sum_{j=1}^{k} \text{var}(D_{i,j})$ is as low as possible given the values $d_{i,j}$ and $r_{i,j}$ for all $i \in \{1, 2\}$ and $j \in \{1, 2, 3, \ldots, k\}$. This holds just when the proportions $\frac{r_{1,j}}{r_j} = \frac{r_{1,j}}{r_{1,j} + r_{2,j}}$ and $\frac{r_{2,j}}{r_j} = \frac{r_{2,j}}{r_{1,j} + r_{2,j}}$ are both constant (and mutually different enough) across all the time points $(t_1, t_2, \ldots, t_k)^T$, and then the log-rank test is the most statistically powerful, i. e. its ability to reject the null hypothesis H_0 claiming the survival curves are equivalent, when they are in fact different, is maximal possible. That is common issue decreasing the test power – the mentioned proportions are typically not constant when a "trend" of the survival curves change a lot, when the curves change their mutual distance or when they even cross themselves one or more times.

B. Principles, assumptions, and limitations of Cox proportional hazard model

Principles of Cox proportional hazard model. The Cox proportional hazard model is frequently used to model relationships between the hazard function of the event of interest, defined as a probability that a subject experiences the event of interest in a small time interval, given that the individual survived up to the beginning of the interval, and explanatory variables. If one of the explanatory variables is categorical, thus dividing an entire sample into two or more groups, then the Cox proportional hazard model could serve as a method for statistical comparing of more than two groups and their survival curves. The hazard function $h(t)$ depending on explanatory variables as suggested by Cox [19], follows for individual i form

$$\log h(t) = \log h_0(t) + \boldsymbol{\beta}^T \boldsymbol{x_i}, \qquad (2)$$

where $h_0(t)$ is the baseline hazard function, $\boldsymbol{\beta}^T = (\beta_0, \beta_1, \beta_2, \dots)^T$ is a vector of estimated linear coefficients to explanatory variables and $\boldsymbol{x_i} = (1, x_{i,1}, x_{i,2}, \dots)^T$ is a vector of values of the explanatory variables for group i. The formula (2) could be after exponentiation rewritten also as

$$h(t) = h_0(t) e^{\boldsymbol{\beta}^T \boldsymbol{x_i}},$$

by which we can see for two groups 1 and 2 that

$$\frac{h(t \mid x_1)}{h(t \mid x_2)} = \frac{h_0(t) e^{\boldsymbol{\beta}^T \boldsymbol{x_1}}}{h_0(t) e^{\boldsymbol{\beta}^T \boldsymbol{x_2}}} = \frac{e^{\boldsymbol{\beta}^T \boldsymbol{x_1}}}{e^{\boldsymbol{\beta}^T \boldsymbol{x_2}}},$$

thus, the hazard ratio for any two groups 1 and 2 is forced to be constant, considering the model (2) and a fact that once estimated coefficients $\hat{\boldsymbol{\beta}}^T = (\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2, \dots)^T$ and input data $\boldsymbol{x_i} = (1, x_{i,1}, x_{i,2}, \dots)^T$ are given, therefore constant. The parameters in the Cox model (2) can be estimated by a partial likelihood [20].

When exists $j \in \{1, 2, 3, \dots\}$ so that β_j is a linear coefficient of a categorical variable classifying observations into two or more groups (with their survival curves), then one can consider the Cox approach as an alternative for the log-rank test with the exception there are more than two survival curves to be compared. Wald t-tests indicate significant statistical differences between the categorical variable levels, thus also in groups' survival curves.

Assumptions and limitations of Cox proportional hazard model. However, while Cox's regression is widely used for event prediction in survival analysis or for comparing more than two survival curves, it has rigid statistical limitations [21]. Particularly, Cox's model assumes that ratios of hazards for any two subjects (individuals or groups) are constant across all time points; that is why the model is called "proportional hazard". However, real survival data often violate this assumption. For instance, supposing two survival curves for two groups as in Fig. 2, such that the curves cross each other, their hazards could not be proportionally constant. Even more, when one of the curves drops to zero while the other levels off similarly to Fig. 3, also, the ratio of the hazard functions could not be constant.

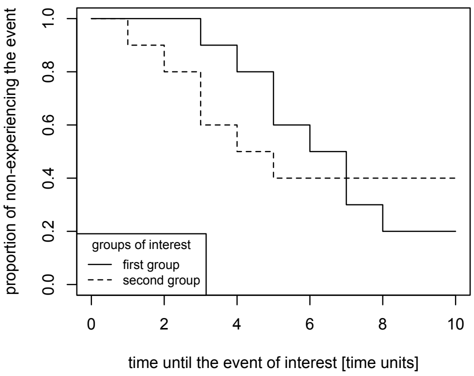


Fig. 2. An example of a pair of survival curves crossing each other.

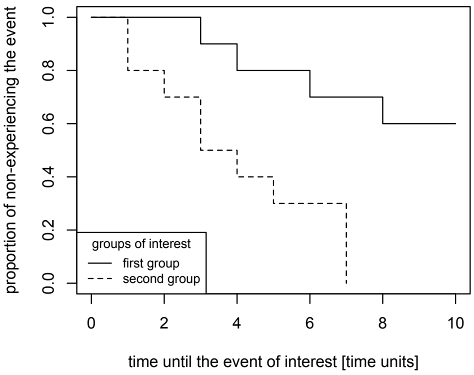


Fig. 3. An example of a pair of survival curves so that one drops to zero while the other levels off.

C. Principles of the random forests

Before we conclude up basic principles of the random forests' algorithm, we remind fundamental pieces of knowledge about decision trees that build up a random forest model.

Principles of the decision trees. The decision trees (also called classification trees) that belong to the CART family of trees (classification and regression trees) are sets of rules that partition the hyperspace of all explanatory variables into disjunctive hyper-rectangles and fit simple (constant) models there, each time minimizing a given criterion [22].

More specifically, the decision trees classify an observation depicted by a vector of values $\boldsymbol{x_i} = (x_{i,1}, x_{i,2}, \dots, x_{i,k})^T$ for k explanatory variables into one of m target classes, i. e. classes of a response categorical variable, where $[k, m] \in \mathbb{N}^2$.

The logic behind a tree induction is described by the flowchart in Fig. 4. Initially, one root node is set, and the tree induction algorithm searches for a node decision rule, i. e. such an explanatory variable and a logical formula containing the explanatory (splitting) variable and its relationship to some constant or subset that minimizes a given criterion. When the optimal node rule is found, the node rule enables to split (binary partition) the dataset into two parts following the logic of the slitting variable and splitting point (the first part contains values larger than or equal to the splitting point, the other contains the rest of dataset). Two new child nodes for the corresponding two parts of the dataset are added to the growing tree. The procedure of searching a node rule, i. e. a splitting variable and splitting point, is repeated for each fresh added (child) node until the part of the dataset that is logically constrained by a set of decision rules coming from the root node till the last (leaf) one, includes observations of only one target class. This strategy of the tree growing is called a <u>t</u>op-<u>d</u>own <u>i</u>nduction of a <u>d</u>ecision <u>t</u>ree (TDIDT).

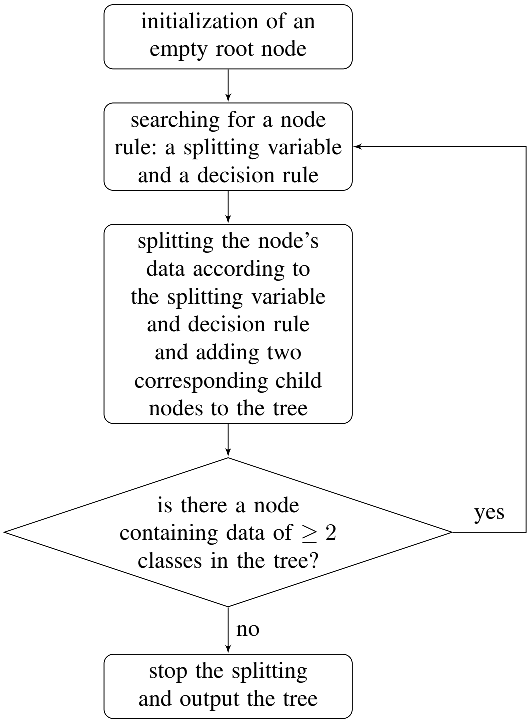

initialization of an empty root node
searching for a node rule: a splitting variable and a decision rule
splitting the node's data according to the splitting variable and decision rule and adding two corresponding child nodes to the tree
is there a node containing data of ≥ 2 classes in the tree?
yes
no
stop the splitting and output the tree

Fig. 4. A top-down induction of a decision tree (TDIDT).

Let $\sigma(\bullet)_j$ be a proportion of a target class j in all observations constrained by rules coming from the root till the node n_t. If the node n_t is a leaf one, it classifies into the class j^* so that $j^* = \text{argmax}_{j \in \{1,2,\ldots,m\}} \{\sigma(\bullet)_j\}$.

The given criterion minimized in searching for node n_t rule is an *impurity measure*, $Q_{n_t}(T)$, such as misclassification error

$$Q_{n_t}(T) = 1 - \sigma(\bullet)_j,$$

or Gini index

$$Q_{n_t}(T) = \sum_{j=1}^{m} \sigma(\bullet)_j (1 - \sigma(\bullet)_j),$$

or deviance (cross-entropy)

$$Q_{n_t}(T) = -\sum_{j=1}^{m} \sigma(\bullet)_j \cdot \log \sigma(\bullet)_j.$$

We can easily see that the higher the $\sigma(\bullet)_j$ as a proportion of a target class j in the node n_t is, the lower whatever kind of the named impurity measures is, as expected.

Following the logic of the top-down induction of a decision tree depicted in Fig. 4, a final tree cannot have lower than maximal possible complexity; even a leaf node including only two observations of two different target classes is once more split into two child leaf nodes. To overcome this issue, *overfitting*, besides some naive approaches like a fixed maximal number of nodes per a tree, etc., a procedure called *pruning* is frequently applied. The pruning is based on numerical estimating of the statistics *cost–complexity function* following the form

$$C_\kappa(T) = \sum_{n_t \in \{\boldsymbol{n}_t\}} |\{\boldsymbol{x}_{n_t}\}| \cdot Q_{n_t}(T) + \kappa \cdot |\{\boldsymbol{n}_t\}|, \qquad (3)$$

where $\{\boldsymbol{n}_t\}$ is a set of leaf nodes of the tree and $\{\boldsymbol{x}_{n_t}\}$ is a set of all observations constrained by rules coming from the root till the node n_t. The idea is to find a subtree T_κ so that $T_\kappa \subset T$ for a given κ that minimizes the statistics $C_\kappa(T)$, i. e. $T_\kappa = \text{argmin}_T \left\{ \sum_{n_t \in \{\boldsymbol{n}_t\}} |\{\boldsymbol{x}_{n_t}\}| \cdot Q_{n_t}(T) + \kappa \cdot |\{\boldsymbol{n}_t\}| \right\}$. The $\kappa \geq 0$ is a hyperparameter (a tuning parameter) and governs the trade-off between a tree complexity or size (low values of κ) and goodness of fit to the data (large values of κ).

Principles of the random forests. Once we can generate classification trees as described above, construction of a random forest is relatively easy. Random forests are finite sets of (distinct) decision trees so that each tree classifies an observation depicted by a vector of values $\boldsymbol{x_i} = (x_{i,1}, x_{i,2}, \ldots, x_{i,k})^T$ for k explanatory variables into one of $m < \infty$ target classes [18]. The eventual classification into the final class is done using a voting scheme – the final class $j^* \in \{1, 2, \ldots, m\}$ is the one that a subset of the random forest's trees classifying just into the class j^* is the largest one among all subsets of the random forest's trees. More technically, $j^* = \text{argmax}_{j \in \{1,2,\ldots,m\}} \{\#\text{ of trees classifying into the class } j\}$. In case of a tie, i. e. there are two or more target classes the forest's trees would classify with maximum frequency into, one of them is picked randomly.

A bit different in the random forest's tree induction is the fact that only $k^* < k$ variables are considered as possible splitting variables in each searching for the node rule. Instead, the subset of k^* variables of the original set of all k explanatory variables is selected randomly using bootstrapping to ensure the pre-selected k^* variables are as much uncorrelated as possible. Other details of the trees inductions are the same as described above. A flowchart of the random forest model building is in Fig. 5.

Assumptions and limitations of the trees and forests. There are neither other technical assumptions nor limitations of the random forests usage worth to be discussed.

III. The proposed method for survival curves comparing

We introduce the novel method for statistical comparison of two or more time-to-event developments of individuals' groups, depicted by their survival curves.

Firstly, data that are on the input of the method have to be transformed. Each individual is originally described

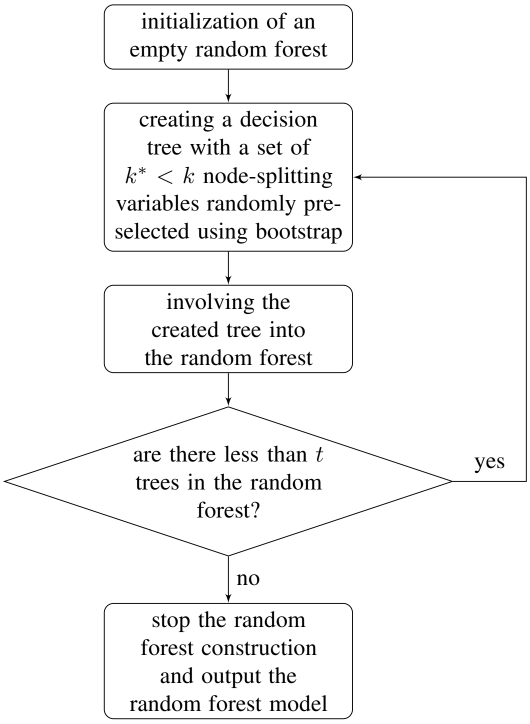

initialization of an empty random forest
creating a decision tree with a set of $k^* < k$ node-splitting variables randomly pre-selected using bootstrap
involving the created tree into the random forest
are there less than t trees in the random forest?
yes
no
stop the random forest construction and output the random forest model

Fig. 5. A construction of the random forest model involving t decision trees.

using their group affiliation, a time to event of interest (or to censoring), and whether they experienced the event of interest (or have been censored). Then, using the original data, for each individual, a sequence of weighted point estimates of probabilities that they did not experience the event of interest in a given time point and the group affiliation is created. That enables introducing new variables (their number is equal to the number or all considered time points) that are used as splitting variables in tree inductions when a random forest model is built.

Once the data are transformed, a random forest model is constructed. Each tree of the random forest either can classify into two or more classes that are represented as the group affiliations, or cannot to classify into any classes at all (then it is necessarily a root node tree), based on its complexity (size). See also Fig. 6.

The more trees of sufficient complexity able to classify into the classes (equal to the groups of individuals, described by their survival curves melted into the transformed variables as mentioned above) are in the forest, the more likely we can reject the null hypothesis that there is no difference between the survival curves (or the groups of individuals' time-to-event development). Thus, a proportion of trees that classify into all the classes, to all the trees of the random forest is very close to a point estimate of the p-value, i. e. the probability we incorrectly reject the null hypothesis of no difference between the survival curves, assuming the null hypothesis is true. Thus, the p-value is a probability of a wrong decision and should be as low as possible whenever we consider rejecting the null hypothesis. The first type error rate, i. e. the incorrect rejection of the null hypothesis when it is true, can be controlled by setting the parameter κ of the random forest's tree complexity (or the tree pruning).

So, the proposed method fulfills all feasible demands on inference testing. We also discuss some of the method's properties, particularly its asymptotic time complexity. The first type error rate is simulated in the simulation study with varying κ tuning parameters. The introduced method is able to compare more than two survival curves, and since it utilizes a random forest tree-based algorithm, it is practically assumption-free. This is where it surpasses both the log-rank test and Cox's regression.

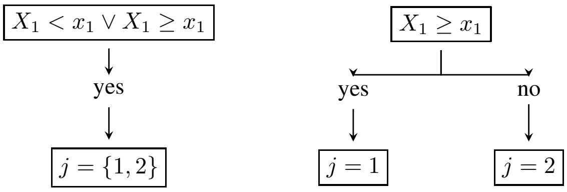


Fig. 6. An example of a root node tree (on the left) not capable to classify into any class unambiguously, and an example of a tree with "sufficient" complexity (on the right) able to classify into two classes ($j = 1$ and $j = 2$).

A. Data transformation and preparation for random forest model building

Initial time-to-event survival data includes n observations; for each of them, we have a piece of information about the time to the event of interest (or to the censoring) and whether the event of interest or the censoring occurred. By adopting the mathematical notation from the section about the log-rank test, for each considered time point t_j, where $j \in \{1, 2, \ldots, k\}$ and $k \in \mathbb{N}$, we can calculate for the group i, where $i \in \{1, 2, \ldots, m\}$, a proportion $r_{i,j}$ of individuals that are at risk (of the event of interest or the censoring) in the j-th time point. Similarly, one can estimate a for the group i, where $i \in \{1, 2, \ldots, m\}$, a proportion $d_{i,j}$ of individuals who experienced the event of interest (or the censoring) in the j-th time point. Putting those estimates together, for the group i, where $i \in \{1, 2, \ldots, m\}$, we can make a point estimate of a probability $\hat{p}_{i,j}$ that an individual from the group would not experience the event of interest (or the censoring) in the j-th time point, so

$$\hat{p}_{i,j} = 1 - \frac{d_{i,j}}{r_{i,j}}. \tag{4}$$

Such an estimate is made k-times for all time points $\{t_1, t_2, \ldots, t_k\}$, by getting $(\hat{p}_{i,1}, \hat{p}_{i,2}, \ldots, \hat{p}_{i,k})$, but such a vector of values is common for all individuals of the group i. However, it could be personalized using an operator $\delta_{\nu,j}$ for ν-th individual, where $\nu \in \{1, 2, \ldots, n\}$, following the form

$$\delta_{\nu,j} = \begin{cases} 1, & \nu\text{-th individual did not experience the event} \\ & \text{of interest in } j\text{-th time point} \\ 0, & \nu\text{-th individual experienced the event} \\ & \text{of interest in } j\text{-th time point,} \end{cases}$$

assuming that ν-th individual belongs to the group i. So by modifying the formula (4) using the operator $\delta_{\nu,j}$ we get

$$\delta_{\nu,j}\hat{p}_{i,j} = \delta_{\nu,j}\left(1 - \frac{d_{i,j}}{r_{i,j}}\right). \tag{5}$$

The logic of the formula (5) enables to get mutually distinct vectors of values $(\delta_{\nu,1}\hat{p}_{i,1}, \delta_{\nu,2}\hat{p}_{i,2}, \dots, \delta_{\nu,k}\hat{p}_{i,k})^T$ for each individual in the group i, which increases natural variability of the data.

Finally, still assuming that ν-th individual belongs to the group i, where $\nu \in \{1, 2, \dots, n\}$, there are n new vectors $(\delta_{\nu,1}\hat{p}_{i,1}, \delta_{\nu,2}\hat{p}_{i,2}, \dots, \delta_{\nu,k}\hat{p}_{i,k})^T$ that could be arranged in a matrix of n rows and k columns, which creates a new dataset suitable as an input for the decision tree induction; the k variables could serve as possible splitting variables in the trees' nodes. The j-th variable of the dataset could be interpreted as a personalized point estimate of probability of non-experiencing the event of interest. The $(k+1)$-th variable in the dataset is a target one – categorical variable describing a group affiliation $i \in \{1, 2, \dots, m\}$ of each observation[2].

B. Construction of the random forest model behind the novel method

The random forest model is built following the algorithms sketched in Fig. 4 and Fig. 5. Variables used as node splitting variables come from the newly created dataset, containing k "explanatory" variables and a target one, as described more in the previous subsection.

Number $t \in \mathbb{N}$ as a count of the trees in the random forest as well as the level of the trees' pruning determined by parameter $\kappa \geq 0$ may vary, as is more explained later.

C. Statistical inference behind the novel method

As already mentioned, the main purpose of the introduced method is to statistically compare two or more survival curves depicting a time-to-event development of distinct groups of individuals. Intuitively, when a large number of the (adequately pruned) trees involved in the random forest model is able to classify into two or more classes, i. e. groups determined by their survival curves, then it is hard to suppose the groups and their survival curves are (statistically) without any difference.

Similarly to the log-rank test or the Cox's regression, let the null hypothesis H_0 claim that there is no statistical difference between the $m > 1$ survival curves[3], and let the alternative hypothesis H_1 claim the contradiction, so

H_0 : *No statistical difference between the m survival curves.*

H_1 : *Statistical difference between the m survival curves.*

Whenever the log-rank test or the Cox's model based on Wald t-test rejects – based on test statistics – the appropriate null hypothesis H_0 in favor of the alternative hypothesis H_1, is equivalent to a situation the test's p-value is lower than or equal to an apriori set level of significance α, usually equal to 0.05. Since the introduced method is in practice assumption-free and non-parametric, the only way to evaluate the statistical inference about the null hypothesis is to estimate the p-value and compare it to the previously set significance level α.

By definition, the p-value is a probability of gaining data evidence at least as extreme as the data evidence actually observed, under the assumption the null hypothesis is true. Let t_c be a number of trees in the random forest that are in contradiction to the null hypothesis under the null hypothesis. The random forest contains exactly t trees. We can easily realize that, given the value for the κ parameter, the value of t_c is equal to the number of all the trees classifying into more than only one class (which is naturally in contradiction to the null hypothesis). Let the $n_c(\tau)$ be a number of classes the tree τ classifies into. Then we can derive

$$t_c = |\{\forall \text{ tree} \in \text{random forest} : n_c(\text{tree}) \geq 2.\}|$$

Then, assuming all trees are inducted randomly regardless of their complexity, the p-value is estimated by $\hat{p}$ so that

$$\begin{aligned}
\hat{p} &= P(\text{getting data at least as extreme as the observed} \mid H_0) = \\
&= P(|\{\forall \text{ tree} \in \text{random forest} : n_c(\text{tree}) \geq 2\}| \geq t_c \mid H_0) = \\
&= P(|\{t_c, t_c+1, \dots, t\}| \mid H_0) = \\
&= \frac{|\{t_c, t_c+1, \dots, t\}|}{t} = \\
&= \frac{t - t_c + 1}{t} = \\
&= 1 - \frac{t_c - 1}{t}.
\end{aligned} \tag{6}$$

Thus, from the formula (6) results that the p-value's estimate is equal to the fraction of $1 - \frac{t_c-1}{t}$. That result is also intuitive. If the initial number t_c of trees in the random forest that are complex enough and classify into two or more classes (and more groups with their survival curves) is in general low, then such a random forest as an entire model is not "so much" in contradiction to the null hypothesis, claiming there are no differences between the classes (and survival curves). Finishing the idea, since the t_c is relatively low, then the fraction $p\text{-value} = 1 - \frac{t_c-1}{t}$ is relatively large, close to 1 and unlikely to be lower than $\alpha(= 0.05)$ which is required for the null hypothesis rejection. On the other hand, when the initial value of t_c is large, i. e. there are many trees in the forest with sufficient complexity classifying into two or more classes (and thus, standing against the null hypothesis), then – because of the large value of t_c – the fraction $p\text{-value} = 1 - \frac{t_c-1}{t}$ is relatively low and likely below the level α. That likely results in the null hypothesis rejection.

The number of trees t in the random forest determines maximum decimal precision of the p-value estimate. When the precision of d decimal digits is required for the p-value estimate, then t has to be $t > 10^d$ or better $t > 10^{d+1}$ to ensure the next-to-last digit (as the d-th decimal digit) is feasibly estimated.

[2] Make a note that across the entire paper, the mathematical notation is consistent – there are m groups depicted by their survival curves, but there are also m target classes of decision trees. Furthermore, there are k time points, and for each of them, a new variable is created within the transformation to the new dataset, thus containing k variables. Finally, the bootstrap behind the random forest model construction also pre-selects $k^* < k$ node variables.

[3] Two or more survival curves; in general $m \in \{2, 3, 4, \dots\}$ curves.

The κ parameter determines how complex the trees in the random forest would be, i. e. how significant the pruning of the trees should be. Inspecting the formula (3), we can simply realize that if $\kappa = 0$, then there is no cost for large tree complexity and the trees in the random forest are generally very complex (of large size). Then, whenever there are at least two observations in the transformed dataset so the they are assigned to different two groups, all the trees (because of the unlimited complexity) in the forest would classify those observations into their groups (classes), i. e. that for each tree τ is $n_c(\tau) \geq 2$, which results into the equity $t_c = t$ and, thus, p-value estimate of p-value $= 1 - \frac{t_c - 1}{t} = 1 - \frac{t-1}{t} = \frac{1}{t} \approx 0$. If p-value ≈ 0, then also p-value $\approx 0 < \alpha$ which, consequently, tends to rejection of the null hypothesis, very likely a *false* rejection that increases the first error type rate. However, high chance of the null hypothesis rejection means also the high statistical power, i. e. the rejection of the null hypothesis when this is not true.

If $\kappa > 0$, then in general the trees' complexity (size) decreases and also not all of the trees are complex enough to classify into more than one class (the are only root node trees); this means that there are trees τ in the random forest so that $n_c(\tau) \leq 1$, and, finally, $t_c < t$. So, p-value estimate is p-value $= 1 - \frac{t_c - 1}{t} > 0$ and it could be, but also could not be below α.

To conclude this, low values of κ tend to decrease values of p-value and increase the statistical power and the first type error rate, and vice versa. However, the more exact relationship between κ and α could be only roughly estimated using simulations due to the stochastic character of the random forests.

D. A brief asymptotic time complexity analysis and fundamental approaches on the p-value estimation

An atomic unit of the random forest model is a decision tree, inducted following the flowchart 4 and algorithm 1. As long as there is a node containing data of ≥ 2 classes, constrained by all node rules coming from root to the node, the data splitting and growing of the tree continues. If the classes in the data are well balanced as well as the growing tree, the splitting partitions the subdatasets roughly in halves, and the average depth of the tree would be $\log n$ and the time complexity would be $\Theta(\log n)$, assuming one split of a node takes 1 time unit. However, on the other hand, when the classes are not well balanced across the dataset, the splitting cuts the subdatasets into 1 and $n - 1$ observations, which takes n steps in total and the depth of the tree is n. Consequently, the asymptotic time complexity is $\Theta(n)$, assuming one split of a node takes 1 time unit.

Within each node splitting, both for a splitting variable among k variables and through the sample size n is searched, the time complexity $\Theta(\bullet)$ of a decision tree building is somewhere in between being in $\Theta(k \cdot n \cdot \log n)$ (the best-case scenario) and $\Theta(k \cdot n \cdot n)$ (the worst-case scenario), so that

$$\Theta(kn \log n) \leq \Theta(\bullet) \leq \Theta(kn^2).$$

Algorithm 1: The top-down induction of decision trees (TDIDT) following the logic of the flowchart 4

```
Data: a $n \times (k+1)$ dataset with transformed variables
Result: a decision tree
$T = (\{\boldsymbol{n}\})$          // a tree $T$ with a set ;
                          // of nodes $\boldsymbol{n}$;
$\{\boldsymbol{n}\} = \{\text{root}\}$    // initially, the tree $T$
 ;
                          // is a root;
$\sigma(\bullet)_j$              // a node criterion;
while $\exists$ a node $\in \{\boldsymbol{n}\}$ so that data constrained by all
  node rules coming from root to this node belong to
  $\geq 2$ classes do
    find for the node a splitting variable and splitting
      point minimizing the $\sigma(\bullet)_j$;
    add to the node two child nodes $n_1$ a $n_2$;
    $\{\boldsymbol{n}\} := \{\boldsymbol{n} \cup \{n_1, n_2\}\}$ ;
    $T := (\{\boldsymbol{n}\})$ // update the tree using
      the new node set $\boldsymbol{n}$ ;
end
a completely inducted tree $T$;
```

When a random forest model containing t trees is constructed, the tree induction as introduced above is repeated t times. That being said, the asymptotic time complexity $\Theta(\bullet\bullet)$ of a random forest model building is in between

$$\Theta(tkn \log n) \leq \Theta(\bullet\bullet) \leq \Theta(tkn^2). \tag{7}$$

One model of the random forest provides one (point) estimate of the p-value using the formula (6). In comparison, the estimation of the χ^2 statistics using the formula (1) takes only $\Theta(2k+1)$ time units since is based on a ratio of two summations of k elements. Fortunately, the time complexity (7) is still polynomial. Furthermore, the building of the random forest with the complexity of (7) could be parallelized; then, asymptotic memory complexity rather than the time complexity could become an issue. In theory, if the random forest building would be parallelized into $\ell \leq t$ independent slave processes each inducting a bunch of $\frac{t}{\ell}$ trees, the time complexity (7) would be reduced to $\Theta\left(\frac{t}{\ell} kn \log n\right) \leq \Theta(\bullet\bullet) \leq \Theta\left(\frac{t}{\ell} kn^2\right)$. Eventually, for $\ell = t$, the random forest building could take the same computing time as only one single tree induction,

$$\begin{gathered}
\Theta\left(\frac{t}{\ell} kn \log n\right) \leq \Theta(\bullet\bullet) \leq \Theta\left(\frac{t}{\ell} kn^2\right) \\
\Theta\left(\frac{t}{t} kn \log n\right) \leq \Theta(\bullet\bullet) \leq \Theta\left(\frac{t}{t} kn^2\right) \\
\Theta\left(kn \log n\right) \leq \Theta(\bullet\bullet) \leq \Theta\left(kn^2\right).
\end{gathered}$$

When we want to estimate the p-value rather using a confidence interval than only using a point, we need to repeat the random forest building many times, let us say $f \gg 0$ times. As a result, we get a set of random forests that might also be called *a primeval superforest of random forests*. The primeval superforest of random forests construction is of a time complexity $\Theta(\bullet\bullet\bullet)$, so that

$$\Theta(ftkn \log n) \leq \Theta(\bullet\bullet\bullet) \leq \Theta(ftkn^2). \tag{8}$$

However, for a given dataset, a point estimate of the p-value is usually supposed to suffice for purposes of routine statistical inference. The primaveral superforest of random forests of the complexity (8) may be applied rather for experimental reasons when e. g., a posterior distribution of the p-values is about to be investigated.

IV. Simulation study

We compared the log-rank test and the proposed method using several simulations of many pairs of survival curves to get preliminary simulated results, although the method – in contrast to the log-rank test – can compare more than only two survival curves. The curves in pairs were assumed they were not significantly different. We calculated the first type error rates, i. e., rates of false test results that two statistically non-different survival curves are (falsely) detected as different. Also, the lower value of the first type error is, the more robust such a method is. The simulation was repeated for different κ parameter values to illustrate how the value of κ determines the first type error rates.

For generating of the pairs of survival curves, we applied the negatively exponential survival function as follows,

$$s(t) = \rho\left(e^{-\frac{5+\varepsilon}{200}t}\right)$$

where ε is a random white noise term following a standard normal distribution, $\varepsilon \sim \mathcal{N}(0, 1^2)$, and $\rho(\bullet)$ is a function rounding its argument to the nearest multiplier of 0.01 using a half rule, e. g. $\sigma(0.012) = 0.01$, $\sigma(0.350) = 0.35$ or $\sigma(0.048) = 0.05$. A group of negatively exponential survival functions following the formula $s(t) = \rho\left(e^{-\frac{5+\varepsilon}{200}t}\right)$ is in Fig. 7.

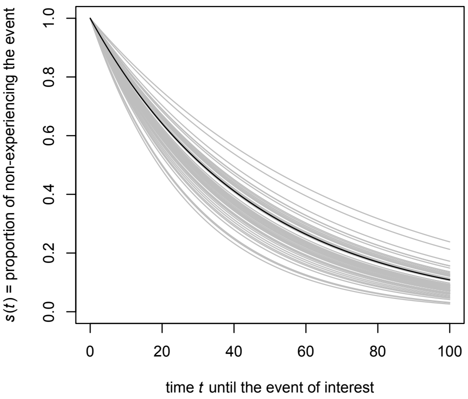


Fig. 7. An example of a group of negatively exponential survival functions following the formula $\rho^{-1}(s(t)) = \rho^{-1}\left(e^{-\frac{5+\varepsilon}{200}t}\right) = e^{-\frac{5+\varepsilon}{200}t}$ for different random values of $\varepsilon \sim \mathcal{N}(0, 1^2)$.

There were $\eta = 1000$ pairs of significantly non-different survival curves generated in total, and for each $\kappa \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, the curves were compared using the log-rank test and the above-proposed method. The number of trees in each random forest was always $t = 1000$. Numbers of cases where p-value was lower than or equal to $\alpha = 0.05$ regardless of the method were summed up, by which we got the point estimates of the first type error rates, as illustrated in table II. The simulation study was performed using R programming language and environment [3]. More on numerical applications of R language to various areas is in [23]–[27].

TABLE II
Point estimates of the first type error rates both for the log-rank test and the proposed method for different values of tuning parameter κ, based on the simulations described above.

	method		
	log-rank test	proposed method	κ
# of simulated cases in total	1000	1000	0.1
# of cases p-value ≤ 0.05	53	65	
first type error rate estimate	0.053	0.065	
# of simulated cases in total	1000	1000	0.3
# of cases p-value ≤ 0.05	48	52	
first type error rate estimate	0.048	0.052	
# of simulated cases in total	1000	1000	0.5
# of cases p-value ≤ 0.05	52	31	
first type error rate estimate	0.052	0.031	
# of simulated cases in total	1000	1000	0.7
# of cases p-value ≤ 0.05	46	14	
first type error rate estimate	0.046	0.014	
# of simulated cases in total	1000	1000	0.9
# of cases p-value ≤ 0.05	55	4	
first type error rate estimate	0.055	0.004	

While the log-rank test returned a point estimate of the first type error rate about 0.050 (regardless of κ since the χ^2 statistics following the formula (1) is not a function of the κ), point estimates of the first type error rates output by the introduced method progressively decreased with increasing value of κ, see table II. What is more, the proposed method seems to be more robust than the log-rank test for large values of κ, based on the simulations above.

V. Conclusion remarks

Survival curves could be compared by the log-rank test when they are only two or by the Cox proportional hazard model if there are more than two curves. However, both methods are limited by statistical assumptions.

We introduced a novel, assumption-free method for survival curves comparison based on a random forest algorithm. Firstly, it requires deriving new variables using the point estimates of modified (personalized) probabilities of non-experiencing the event of interest across all time points. Using those variables as node splitting ones, the random forest model can be built. A subtraction between 1 and a proportion of trees with sufficient complexity, capable of classifying into two or more classes, i. e. groups determined by their survival curves, to all trees of the forest, is a point estimate of p-value of the proposed method. Parameter κ determines the random forest trees' complexity, and, thus, by increasing the parameter, the first type error rate decreases, and robustness of the method increases, as was also illustrated within the simulation study.

The asymptotic time complexity of the random forest-based method is higher than the one for the log-rank test but still polynomial and could be parallelized, too.

The random forest-based method seems to overcome the risk of violations of statistical assumptions of the traditional techniques comparing survival curves and, furthermore, could compare more than two survival curves. Eventually, the method and its computational optimization could also inspire a new R package development.

VI. Acknowledgement

This paper is supported by the grant OP VVV IGA/A, CZ.02.2.69/0.0/0.0/19_073/0016936 with no. 18/2021, which has been provided by the Internal Grant Agency of the Prague University of Economics and Business.

References

[1] E. L. Kaplan and Paul Meier. "Nonparametric Estimation from Incomplete Observations". In: *Journal of the American Statistical Association* 53.282 (June 1958), pp. 457–481. DOI: 10.1080/01621459.1958.10501452. URL: https://doi.org/10.1080/01621459.1958.10501452.

[2] Huimin Li, Dong Han, Yawen Hou, et al. "Statistical Inference Methods for Two Crossing Survival Curves: A Comparison of Methods". In: *PLOS ONE* 10.1 (Jan. 2015). Ed. by Zhongxue Chen, e0116774. DOI: 10.1371/journal.pone.0116774. URL: https://doi.org/10.1371/journal.pone.0116774.

[3] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2017. URL: https://www.R-project.org/.

[4] Therneau T. *survival: A Package for Survival Analysis in R*. Vienna, Austria, R package version 3.1-12. URL: https://CRAN.R-project.org/package=survival/.

[5] F. Kong. "Robust covariate-adjusted logrank tests". In: *Biometrika* 84.4 (Dec. 1997), pp. 847–862. DOI: 10.1093/biomet/84.4.847. URL: https://doi.org/10.1093/biomet/84.4.847.

[6] Rui Song, Michael R. Kosorok, and Jianwen Cai. "Robust Covariate-Adjusted Log-Rank Statistics and Corresponding Sample Size Formula for Recurrent Events Data". In: *Biometrics* 64.3 (Dec. 2007), pp. 741–750. DOI: 10.1111/j.1541-0420.2007.00948.x. URL: https://doi.org/10.1111/j.1541-0420.2007.00948.x.

[7] Richard Peto and Julian Peto. "Asymptotically Efficient Rank Invariant Test Procedures". In: *Journal of the Royal Statistical Society. Series A (General)* 135.2 (1972), p. 185. DOI: 10.2307/2344317. URL: https://doi.org/10.2307/2344317.

[8] Georg Heinze, Michael Gnant, and Michael Schemper. "Exact Log-Rank Tests for Unequal Follow-Up". In: *Biometrics* 59.4 (Dec. 2003), pp. 1151–1157. DOI: 10.1111/j.0006-341x.2003.00132.x. URL: https://doi.org/10.1111/j.0006-341x.2003.00132.x.

[9] Song Yang and Ross Prentice. "Improved Logrank-Type Tests for Survival Data Using Adaptive Weights". In: *Biometrics* 66.1 (Apr. 2009), pp. 30–38. DOI: 10.1111/j.1541-0420.2009.01243.x. URL: https://doi.org/10.1111/j.1541-0420.2009.01243.x.

[10] Chenxi Li. "Doubly robust weighted log-rank tests and Renyi-type tests under non-random treatment assignment and dependent censoring". In: *Statistical Methods in Medical Research* 28.9 (July 2018), pp. 2649–2664. DOI: 10.1177/0962280218785926. URL: https://doi.org/10.1177/0962280218785926.

[11] Donald G. Thomas. "Exact and asymptotic methods for the combination of 2 × 2 tables". In: *Computers and Biomedical Research* 8.5 (Oct. 1975), pp. 423–446. DOI: 10.1016/0010-4809(75)90048-8. URL: https://doi.org/10.1016/0010-4809(75)90048-8.

[12] Cyrus R. Mehta, Nitin R. Patel, and Robert Gray. "Computing an Exact Confidence Interval for the Common Odds Ratio in Several 2 × 2 Contingency Tables". In: *Journal of the American Statistical Association* 80.392 (Dec. 1985), p. 969. DOI: 10.2307/2288562. URL: https://doi.org/10.2307/2288562.

[13] Lubomír Štěpánek, Filip Habarta, Ivana Malá, et al. "Analysis of asymptotic time complexity of an assumption-free alternative to the log-rank test". In: *Proceedings of the 2020 Federated Conference on Computer Science and Information Systems*. IEEE, Sept. 2020. DOI: 10.15439/2020f198. URL: https://doi.org/10.15439/2020f198.

[14] Karl Mosler. *Multivariate dispersion, central regions, and depth : the lift zonoid approach*. New York: Springer, 2002. ISBN: 0387954120.

[15] Tomasz Smolinski. *Computational intelligence in biomedicine and bioinformatics : current trends and applications*. Berlin: Springer, 2008. ISBN: 978-3-540-70776-9.

[16] Alexander Kulikov. *Combinatorial pattern matching : 25th annual symposium, CPM 2014 Moscow, Russia, June 16-18, 2014, proceedings*. Cham: Springer, 2014. ISBN: 978-3-319-07565-5.

[17] Nihal Ata Tutkun and Muhammet Tekin. "Cox Regression Models with Nonproportional Hazards Applied to Lung Cancer Survival Data". In: *Hacettepe Journal of Mathematics and Statistics Volume* 36 (Jan. 2007), pp. 157–167.

[18] Leo Breiman. "Random Forests". In: *Machine Learning* 45.1 (2001), pp. 5–32. DOI: 10.1023/a:1010933404324. URL: https://doi.org/10.1023/a:1010933404324.

[19] D. R. Cox. "Regression Models and Life-Tables". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 34.2 (1972), pp. 187–220. ISSN: 00359246. URL: http://www.jstor.org/stable/2985181.

[20] Lubomír Štěpánek, Filip Habarta, Ivana Malá, et al. "A Machine-learning Approach to Survival Time-event Predicting: Initial Analyses using Stomach Cancer Data". In: *2020 International Conference*

on e-Health and Bioengineering (EHB). IEEE, Oct. 2020. DOI: 10.1109/ehb50910.2020.9280301. URL: https://doi.org/10.1109/ehb50910.2020.9280301.

[21] Xiaonan Xue, Xianhong Xie, Marc Gunter, et al. "Testing the proportional hazards assumption in case-cohort analysis". In: *BMC Medical Research Methodology* 13.1 (July 2013). DOI: 10.1186/1471-2288-13-88. URL: https://doi.org/10.1186/1471-2288-13-88.

[22] Leo Breiman. *Classification and regression trees*. New York: Chapman & Hall, 1993. ISBN: 9780412048418.

[23] Lubomír Štěpánek, Pavel Kasal, and Jan Měšťák. "Evaluation of facial attractiveness for purposes of plastic surgery using machine-learning methods and image analysis". In: *2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom)*. IEEE, Sept. 2018. DOI: 10.1109/healthcom.2018.8531195. URL: https://doi.org/10.1109/healthcom.2018.8531195.

[24] Lubomír Štěpánek, Pavel Kasal, and Jan Měšťák. "Machine-learning at the service of plastic surgery: a case study evaluating facial attractiveness and emotions using R language". In: *Proceedings of the 2019 Federated Conference on Computer Science and Information Systems*. IEEE, Sept. 2019. DOI: 10.15439/2019f264. URL: https://doi.org/10.15439/2019f264.

[25] Lubomír Štěpánek, Pavel Kasal, and Jan Měšťák. "Machine-Learning and R in Plastic Surgery – Evaluation of Facial Attractiveness and Classification of Facial Emotions". In: *Advances in Intelligent Systems and Computing*. Springer International Publishing, Sept. 2019, pp. 243–252. DOI: 10.1007/978-3-030-30604-5_22. URL: https://doi.org/10.1007/978-3-030-30604-5_22.

[26] Lubomír Štěpánek, Pavel Kasal, and Jan Měšťák. "Machine-learning at the service of plastic surgery: a case study evaluating facial attractiveness and emotions using R language". In: *Proceedings of the 2019 Federated Conference on Computer Science and Information Systems*. IEEE, Sept. 2019. DOI: 10.15439/2019f264. URL: https://doi.org/10.15439/2019f264.

[27] Lubomír Štěpánek, Pavel Kasal, and Jan Měšťák. "Evaluation of Facial Attractiveness after Undergoing Rhinoplasty Using Tree-based and Regression Methods". In: *2019 E-Health and Bioengineering Conference (EHB)*. IEEE, Nov. 2019. DOI: 10.1109/ehb47216.2019.8969932. URL: https://doi.org/10.1109/ehb47216.2019.8969932.

DOI: 10.15439/2021F141
ISSN 2300-5963 ACSIS, Vol. 25

Practical parallelization of Gear-Nordsieck and Brayton-Gustavson-Hatchel stiff ODE solver

Marek Stabrowski
Warsaw University of Technology, Poland, e-mail: marek.2491@gmail.com

***Abstract*—The paper compares two ODE solvers using an example of a heat transfer equation. The sequential version of Brayton-Gustavson-Hatchel solver has been slightly inferior to Gear-Nordsieck solver. Algorithms profiling has led to the decision of parallelizing linear equation solving section and function evaluation. The first approach (parallelizing linear equations) improves performance of both algorithms. Second approach (parallelizing function evaluation) boosts BGH solver performance. Finally, it has been proved that wholly parallel version of BGH solver is more efficient with respect to processing time.**

***Index Terms*—differential equations, Brayton-Gustavson-Hatchel ODE solver, Gear-Nordsieck ODE solver, parallel computations**

I. Introduction

THE PROBLEM of parallelism introduction in the field of ordinary differential equations (ODE) systems is not new. During the last 50 years three main directions of parallel-techniques have been investigated [4] :
- across the method — e.g. independent stages of Runge-Kutta or extrapolation integrators evaluated in parallel;
- across the problem — e.g. waveform relaxation;
- across the time-domain - e.g. PINT, PFASST .
This paper will be devoted to parallelization across the method, applied to the field of stiff ODE solvers.

II. Basic features of Brayton-Gustavson-Hatchel ODE solver

Gear-Nordsieck method [3] is at present a classic tool for solving of stiff ordinary differential equations (ODE). Critical analysis of backward differentiation formulas (BDF) method helped to select the possible challenger - a method developed some 20 years ago by Brayton, Gustavson and Hatchel (in further course BGH method) [5]. A problem to be solved, i.e. ODE system, may be written in the form:

$$f(x, \dot{x}, t) = 0, \qquad 0 \leq t \leq T \tag{1}$$

where f is a vector (a set of functions). The Gear method uses Nordsieck vector components

$$(x_n, h_n, \dot{x_n}, 1/2h_n^2\ddot{x}_n, ..., (1/k!)h_n^k y_n^{(k)}) \tag{2}$$

as basic backward information. Brayton, Gustavson and Hatchel have forwarded the thesis that usage of backward information in the form

$$x_{n-j}, \qquad j = 0, 1, ...k \tag{3}$$

is more efficient and leads to stable formulas, even for rapidly changing step size h.

The implementation of BGH method developed by the author [5] features variable order, operation count has been reduced and new efficient error control algorithm has been introduced. Predictor and corrector coefficients are computed through actualisation of old ones. Antisymmetry of square arrays is taken into account. Two-dimensional arrays are effectively indexed as one-dimensional. New values of step size reduction and expansion coefficients have been introduced. Asymmetric dead space in order changing section helps eliminate unnecessary order thrashing. The results of comparison of BGH algorithm [5] and open source version of Gear-Nordsieck algorithm [3], show competitiveness of BGH algorithm.

III. An example of real world ODE system – heat transfer problem

An example of heat diffusion through the wall will be used for comparison of both algorithms in sequential and parallel versions. The heat conduction equation for this case has the form

$$\frac{\partial T}{\partial t} = \frac{\lambda}{c_p \rho} \frac{\partial^2 T}{\partial x^2} \tag{4}$$

where T - temperature depends on both time and place in the wall, t - time, λ - heat transfer coefficient in the wall material, c_p - concrete heat capacity coefficient, ρ - concrete density, x - coordinate location measured across the wall.

In order to solve this parabolic partial differential equation, the derivative in space can be represented in differential form by dividing the wall thickness L into a finite number of N nodes. A system of ordinary differential equations describing temperature changes over time in individual nodes is then obtained:

$$\frac{dT_j}{dt} = \frac{\lambda}{c_p \rho} \frac{T_{j-1}^n - 2T_j^n + T_{j+1}^n}{(\Delta x)^2} \tag{5}$$

This equation describes temperature changes in nodes located inside the wall. The temperature at the edge nodes (left and right wall surfaces) can be determined from simple algebraic equations averaging the temperature inside and outside the wall.

The heat diffusion through the wall is now described by the N-2 system of first order ordinary differential equations (5) and two algebraic equations. This problem can be easily scaled, i.e. the number of differential equations (5) can be changed.

This work was not supported by any organization

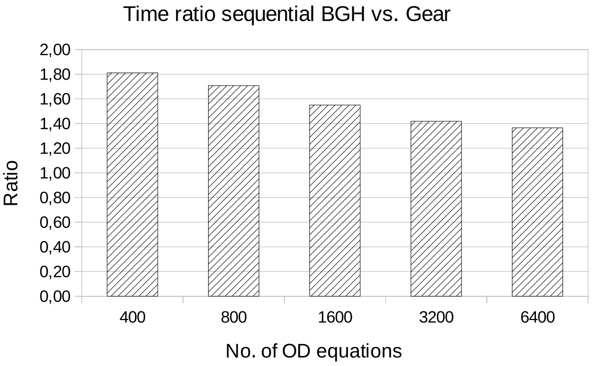


Fig. 1. Execution time ratio of BGH algorithm vs. Gear algorithm - sequential versions

IV. Comparison of Brayton-Gustavson-Hatchel and Gear-Nordsieck sequential ODE solvers

The implementation of Gear-Nordsieck algorithm redesigned by J. P. Moreau [3] has been selected for the comparisons in current research. The tests reported here have been performed on the computer with SkyLake processor. It features four physical cores (threads) and the additional four virtual cores/threads (hyperthreading). The source code of both algorithms has been compiled with C/C++ compiler version 8.3.1 and subsequently has been run on the Skylake desktop (4 physical cores) with Linux Fedora 31 operating system.

Temperature distribution in the concrete wall (thickness = 0.1 m) have been computed for the time points 2, 4, 6 sec. Efficiency of sequential BGH algorithm, in the sense of execution time, is only slightly inferior to Gear algorithm (fig. 1). It can be observed that the advantage of Gear algorithm diminishes with increased number of ordinary differential equations. For 400 equations Gear algorithm outperforms BGH algorithm by the factor of almost 2.0 but for 6400 equations this factor falls to 1.2.

V. Profiling of BGH and Gear ODE solvers

It is advisable, before any form of software tuning, to locate critical sections, functions and subroutines, consuming meaningful execution time. Profiling of both algorithms (for 800 equations) implementations has been carried out with the aid of valgrind/callgrind tool. Two following tables (table I and table II) present representative sample data of sections/subroutines call count and approximate percent share of execution times.

TABLE I
Callgrind profiling of serial BGH algorithm

	approx. %% time	no. of calls
BGHstiff	1.17	8000
solve	95.36	812
decomp	2.52	46
HeatTransfer	0.67	74414
interp	0.11	8000
predictor	0.06	1208

TABLE II
Callgrind profiling of serial Gear-Nordsieck algorithm

	approx. %% time	no. of calls
gear4	6.95	4
awp	5.85	16
engl45	49.1	483351
dgl14	28.95	213
gausol	3.14	8000
gaudec	2.88	71

Subroutines performing linear equations LU decomposition and solving gausol and gaudec are counterparts of decomp and solve. Computation of the function to be integrated is located in subroutines engl45, dgl14 vs. HeatTransfer. It is apparent that the function evaluation (formula (5) for BGH solver) is one candidate for parallelization (see section 6).

Another parallelization candidate is linear equation solving routine gausol and solve. Both these routines consume more execution time than LU decomposition routines. However, such conclusion and approach is superficial and naive. It has been proved elsewhere [6] that the decomposition routines gaudec and decomp are more promising with respect to parallelization (see section 7).

The number of the linear equations routines calls does not depend on the number of nodes, i.e. on the number of differential equations. However, the dimension of the system rises with the square of the nodes number. It is quite reasonable to expect, that parallelization will be more efficient in the case of larger, fine-grained systems. Different results may be expected in the case of parallelization and fine-tuning of function evaluation. In the case of 400 differential equations BGH algorithm performs about 22% of function evaluations with respect to Gear algorithm. For 6400 ordinary differential equations this ratio falls to 1%. It may be expected that BGH algorithm will be more efficient in the case of more complex ODE formulas.

VI. Parallelization of function computation in Gear and BGH ODE solvers

At first the results of parallel computation of ODE function (5) will be presented. Parallelization will be implemented in both cases with the aid of POSIX pthreads library [1, 2]. Computation is performed in N nodes across the wall. Quite naturally, this set of N computations may be divided into the segments assigned to individual cores through creation of appropriate threads.

Parallelization has limited influence on computation efficiency (timing) for 400 equations (fig.2). The situation is better for 800 equations, as forking of 4 threads speeds-up the processing by the factor of 3.5, reaching 6 for 8 threads. For 1600 equations, the speed-up factor reaches the value of 4 for 4 threads and 7 to 8 for 8 threads. Limited speed-up

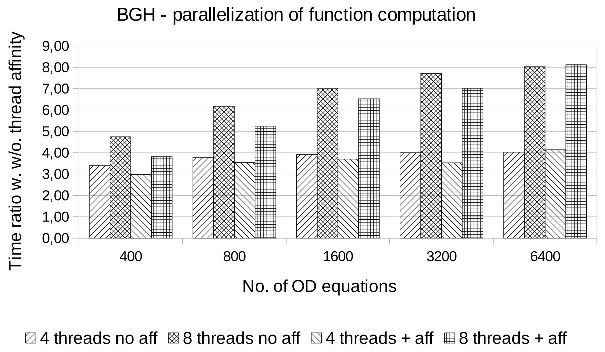


Fig. 2. The effect of function (5) parallelization in BGH algorithm; 4 and 8 threads without and with threads affinity

for lower equations count results from overhead of threads forking. In another series of experiments, an attempt to limit the overhead of threads forking has been implemented. It has been achieved through fixed bounding of individual threads with specific processor cores (affinity mechanism). For lower equation count, introduction of threads affinity has adverse influence on computation efficiency (fig. 2). The setting of threads affinity incurs higher overhead of thread forking. For higher equation count (e.g. at 1600 equations and above) this additional overhead is relatively small, compared with real number crunching inside individual threads. Summing it up - there has been no execution speed-up due to threads affinity setting.

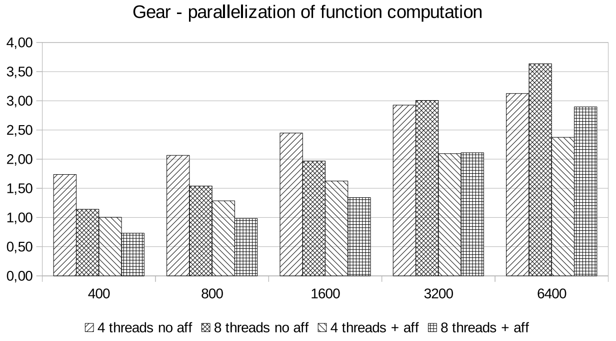


Fig. 3. The effect of function (5) parallelization in Gear algorithm; 4 and 8 threads without and with threads affinity

Similar experiments with parallelization of function evaluation in Gear-Nordsieck algorithm, presented in fig. 3, are disappointing. In the case of basic 4-thread parallelization, the speed-up ranges from 2 (800 equations) up to 3. Hyperthreading improves slightly the results for higher number of equations. Affinity setting degrades the performance. Comparison of BGH and Gear-Nordsieck solvers with parallelized function evaluation seems to confirm the preliminary profiling research. Parallelizing of this code section improves the performance of BGH solver almost by the factor equal to the number of processor cores. In the case of Gear-Nordsieck solver, the improvement is markedly lower, moreover for basic (without affinity) parallelization only.

VII. Parallelization of linear equation solving in Gear and BGH ODE solvers

The second area of the BGH and Gear algorithms, potentially amenable to parallelization and speed-up, is the linear equation solving section. Parallel linear equation solvers have been designed and tested very extensively [6, 2]. The efficiency of parallelization depends, among others, on the sparsity of the coefficient matrix. In general, parallel speed-up is larger in the case of rather dense matrices and falls down for the sparse ones.

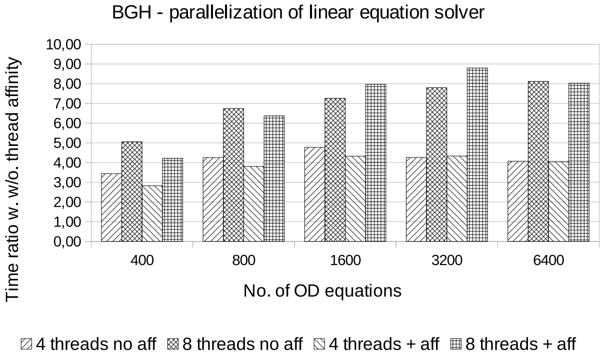


Fig. 4. The effect of linear equation solver parallelization in BGH algorithm; 4 and 8 threads without and with threads affinity

Straightforward parallelization (i.e. without thread affinity) results in speed-up proportional to the number of threads (fig. 4). The results for lowest equation count are slightly inferior, as the parallelization gains are offset by the overhead of threads forking. Similarly, as in the case of function evaluation, introduction of thread affinity does not improve efficiency.

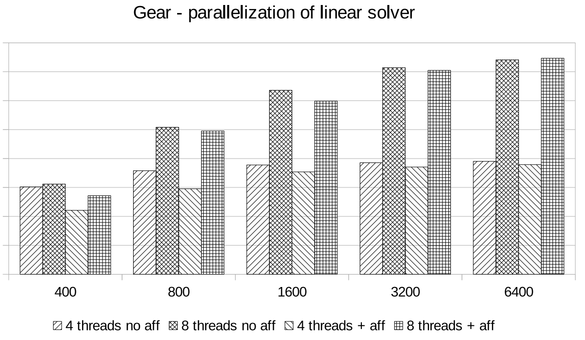


Fig. 5. The effect of linear equation solver parallelization in Gear algorithm; 4 and 8 threads without and with threads affinity

Parallelization of linear equation solver in Gear-Nordsieck algorithm leads to similar results. For higher equations/nodes number, the speed-up (fig. 5) is almost proportional to the number of forked threads. The improvement for lower equation count (800 and below) is lower than in the case of BGH solver. Also, the influence of affinity setting is negligible.

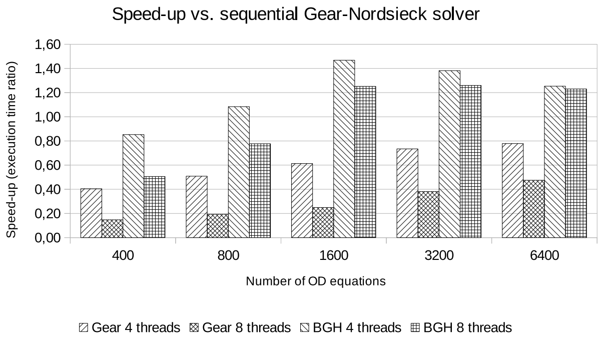


Fig. 7. Cumulative speed-up of multi-threaded solvers with linear equations and function computation vs. sequential Gear-Nordsieck solver

VIII. Comparison of the solvers with parallel function evaluation

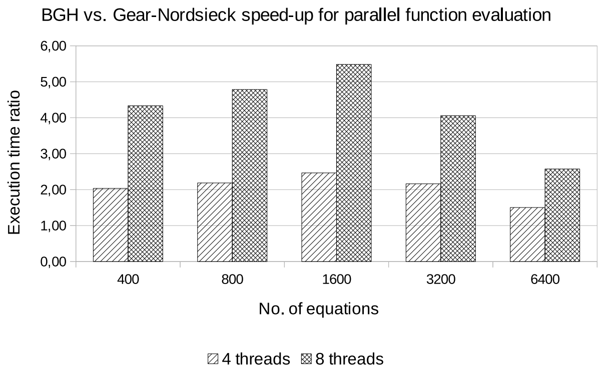


Fig. 6. Comparison of BGH and Gear-Nordsieck solvers with parallel function evaluation; 4 and 8 threads

It has been proved that parallelization of linear equation solving improves the performance of both solvers in almost equal degree with small advantage of BGH solver. Parallelization of function evaluation favors BGH solver. Direct comparison of such parallel versions of both solvers is presented in fig. 6. It follows that BGH solver is faster by the factor of 2 for 4-thread version, reaching the speed-up of 4 to 5 for 8-thread (hyperthreading) version. This advantage is a bit lower for higher equation count.

IX. Cumulative comparison of the solvers

In previous sections, two partial parallelization modifications of both solvers have been presented and investigated. However, the end user is rather interested in the final cumulative effect of these modifications. In order to perform such comparison, a basic sequential Gear-Nordsieck solver has been selected as the reference. Both solvers have been parallelized in a cumulative way, i.e. through parallel linear equation and function computation. First, it can be observed (fig. 7) that hyperthreading leads to inferior performance, as compared with 4-thread version. Next, parallel versions of Gear-Nordsieck solver are significantly slower than the sequential version. Third observation reveals good parallelization potential of BGH solver. Parallel version of BGH solver outperforms the fastest version of Gear-Nordsieck solver by 20-40% for higher equation count.

X. Conclusions

Comparison of basic sequential version of Gear-Nordsieck ODE solver and Brayton-Gustavson-Hatchel solver has shown that both solvers are almost equally efficient with regard to execution time. The performance of both solvers improves with rising number of differential equations. Two most promising sections of both solvers have been parallelized. The improvement of execution time has been observed in the case of linear equation solving parallelization. The speed-up has been proportional to the number of forked threads, at least for higher equation number. Parallelization of function evaluation has led to similar improvement only in the case of BGH solver. The speed-up in the case of Gear solver has been significantly lower than the threads number with limitation to most basic parallelization (without hyperthreading). These results conform to introductory profiling analysis. For both solvers, introduction of thread affinity in both parallelization cases, i.e. equation solving and function computation, has adverse influence or no influence on execution timing.

References

[1] J. Bylina. “A Framework for Generating and Evaluating Parallelized Code”. In: *Proceedings of the 2017 Federated Conference on Computer Science and Information Systems*. Vol. 11. 2017, pp. 493–496. DOI: 10.15439/2017F230.

[2] S. Fialko and V. Karpilovskyi. “Multithreaded Parallelization of the Finite Element Method Algorithms for Solving Physically Nonlinear Problems”. In: *Proceedings of the 2018 Federated Conference on Computer Science and Information Systems*. Vol. 15. 2018, pp. 311–318. DOI: 10.15439/2018F40.

[3] J. P. Moreau. *Website dedicated to numerical analysis*. 2021. URL: http://jean-pierre.moreau.pagesperso-orange.fr.

[4] S. I. Solodushkin and I. F. Iumanova. “Parallel Numerical Methods for Ordinary Differential Equations: a Survey”. In: *CEUR Workshop Proceedings*. Vol. 1729. 2016, pp. 1–10. URL: http://ceur-ws.org/Vol-1729/paper-01.

[5] M. Stabrowski. “Efficient Algorithm for Solving of Stiff Ordinary Differential Equations”. In: *Simulation Practice and Theory* 5 (1997), pp. 333–344. URL: https://www.sciencedirect.com/journal/simulation-modelling-practice-and-theory.

[6] M. Stabrowski. “Parallel Real-world LU Decomposition: Gauss vs Crout Algorithm”. In: *Open Computer Science* (2018), pp. 210–217. URL: https://www.degruyter.com/view/j/comp.

14^{th} International Symposium on Multimedia Applications and Processing

SOFTWARE Engineering Department, Faculty of Automation, Computers and Electronics, University of Craiova, Romania “Multimedia Applications Development” Research Centre

Background and Goals

Multimedia and information have become ubiquitous on the web and communication services, creating new challenges for detection, recognition, indexing, access, search, retrieval, automated understanding, processing and generation of several applications which are using image, signal or various multimedia technologies.

Recent advances in pervasive computers, networks, telecommunications, and information technology, along with the proliferation of multimedia mobile devices—such as laptops, iPods, personal digital assistants (PDA), and smartphones—have stimulated the rapid development of intelligent applications. These key technologies by using Virtual Reality, Augmented Reality and Computational Intelligenceare creating a recent multimedia revolution which will have significant impact across a wide spectrum of consumer, business, healthcare, educational and governmental domains. Yet many challenges remain.

We welcome papers covering innovative applications, practical usage but also theoretical aspects of the above mentioned trends. The key objective of this sessionis to gather results from academia and industry partners working in all subfields of multimedia and language: content design, development, authoring and evaluation, systems/tools oriented research and development. We are also interested in looking at service architectures, protocols, and standards for multimedia communications—including middleware—along with the related security issues. Finally, we encourage submissions describing work on novel applications that exploit the unique set of advantages including home-networked entertainment and games. However, innovative contributions which don’t exactly fit into these areas are also welcomed to this session.

The Multimedia Applications and Processing (MMAP) will provide an opportunity for researchers and professionals to discuss present and future challenges as well as potential collaboration for future progress in the field. The MMAP Symposium welcomes submissions of original papers concerning all aspects of multimedia domain ranging from concepts and theoretical developments to advanced technologies and innovative applications. MMAP invites original previously unpublished contributions that are not submitted concurrently to a journal or another conference. Papers acceptance and publication will be judged based on their relevance to the symposium theme, clarity of presentation, originality and accuracy of results and proposed solutions.

Call for Papers

MMAP 2020 is a major forum for researchers and practitioners from academia, industry, and government to present, discuss, and exchange ideas that address real-world problems with real-world solutions.

The MMAP 2020 Symposium welcomes submissions of original papers concerning all aspects of multimedia domain ranging from concepts and theoretical developments to advanced technologies and innovative applications. MMAP 2020 invites original previously unpublished contributions that are not submitted concurrently to a journal or another conference. Papers acceptance and publication will be judged based on their relevance to the symposium theme, clarity of presentation, originality and accuracy of results and proposed solutions.

Topics

- Audio, Image and Video Processing
- Animation, Virtual Reality, 3D and Stereo Imaging
- Big Data Science and Multimedia Systems
- Cloud Computing and Multimedia Applications
- Machine Learning, Fuzzy Systems, Neural Networks and Computational Intelligence for Information Retrieval in Multimedia Applications
- Data Mining, Warehousing and Knowledge Extraction
- Multimedia File Systems and Databases: Indexing, Recognition and Retrieval
- Multimedia in Internet and Web Based Systems
- E-Learning, E-Commerce and E-Society Applications
- Human Computer Interaction and Interfaces in Multimedia Applications
- Multimedia in Medical Applications and Computational biology
- Entertainment, Personalized Systems and Games
- Security in Multimedia Applications: Authentication and Watermarking
- Distributed Multimedia Systems
- Network and Operating System Support for Multimedia
- Mobile Network Architecture and Fuzzy Logic Systems
- Intelligent Multimedia Network Applications
- Future Trends in Computing System Technologies and Applications
- Trends in Processing Multimedia Information

Best Paper Award

A best paper award will be made for work of high quality presented at the MMAP Symposium. Award comprises a certificate for the authors and will be announced on time of conference. Selected papers will be invited to high IF journals organized for the participants of MMAP.

- Authors should submit draft papers (as Postscript, PDF or MSWord file).
- The total length of a paper should not exceed 10 pages IEEE style (including tables, figures and references). IEEE style templates are available here.
- Papers will be refereed and accepted on the basis of their scientific merit and relevance to the workshop.
- Preprints containing accepted papers will be published on a USB memory stick provided to the FedCSIS participants.
- Only papers presented at the conference will be published in Conference Proceedings and submitted for inclusion in the IEEE Xplore®database.
- Conference proceedings will be published in a volume with ISBN, ISSN and DOI numbers and posted at the conference WWW site.
- Conference proceedings will be indexed in BazEkon and submitted for indexation in: Thomson Reuters—Conference Proceedings Citation Index, SciVerse Scopus, Inspec, Index Copernicus, DBLP Computer Science Bibliography and Google Scholar
- Extended versions of selected papers presented during the conference will be published as Special Issue(s).
- Organizers reserve right to move accepted papers between FedCSIS events.

Advisory Board

- **Neustein, Amy,** Boston University, USA, Editor of Speech Technology
- **Jain, Lakhmi C.,** University of South Australia and University of Canberra, Australia
- **Zurada, Jacek,** University of Louisville, United States
- **Ioannis, Pitas,** University of Thessaloniki, Greece
- **Badica, Costin,** University of Craiova, Romania
- **Borko, Furht,** Florida Atlantic University, USA
- **Kosch, Harald,** University of Passau, Germany
- **Uskov, Vladimir,** Bradley University, USA
- **Deserno, Thomas M.,** Aachen University, Germany
- **Burdescu, Dumitru Dan,** University of Craiova, Romania

Technical Session Chair

- **Korzhik, Valery,** State University of Telecommunications, Russia
- **Schiopoiu Burlea, Adriana,** University of Craiova, Romania

Program Committee

- **Azevedo, Ana,** CEOS.PP-ISCAP/IPP, Portugal
- **Badica, Amelia,** University of Craiova, Romania
- **Cano, Alberto,** Virginia Commonwealth University, United States
- **Cordeiro, Jose,** EST Setúbal/I.P.S.
- **Cretu, Vladimir,** Politehnica University of Timisoara, Romania
- **Debono, Carl James,** University of Malta, Malta
- **Fabijańska, Anna,** Lodz University of Technology, Poland - Institute of Applied Computer Science, Poland
- **Fomichov, Vladimir,** National Research University Higher School of Economics, Moscow, Russia., Russia
- **Giurca, Adrian,** Brandenburg University of Technology, Germany
- **Grosu, Daniel,** Wayne State University, United States
- **Kabranov, Ognian,** Cisco Systems, United States
- **Keswani, Dr. Bright,** Suresh Gyan Vihar University, Mahal, Jagatpura, Jaipur
- **Kotenko, Igor,** St. Petersburg Institute for Informatics and Automation of the Russian Academy of Science, Russia
- **Logofatu, Bogdan,** University of Bucharest, Romania
- **Mangioni, Giuseppe,** DIEEI - University of Catania, Italy
- **Marghitu, Daniela,** Auburn University
- **Mihaescu, Cristian,** University of Craiova, Reunion
- **Mocanu, Mihai,** University of Craiova, Romania
- **Ohzeki, Kazuo,** Professor Emeritus at Shibaura Institute of Technology, Japan
- **Pohl, Daniel,** Intel, Germany
- **Popescu, Dan,** CSIRO, Sydney, Australia, Australia
- **Popescu, Daniela E.,** Integrated IT Management Service, University of Oradea
- **Querini, Marco,** Department of Civil Engineering and Computer Science Engineering
- **Radulescu, Florin,** University "Politehnica" of Bucharest
- **Romansky, Radi,** Professor at Technical University of Sofia
- **Rutkauskiene, Danguole,** Kaunas University of Technology
- **Salem, Abdel-Badeeh M.,** Ain Shams University, Egypt
- **Sari, Riri Fitri,** University of Indonesia, Indonesia
- **Scherer, Rafał,** Częstochowa University of Technology, Poland
- **Sousa Pinto, Agostinho,** Instituto Politécnico do Porto, Portugal
- **Stanescu, Liana,** University of Craiova, Romania
- **Stoicu-Tivadar, Vasile,** University Politehnica Timisoara
- **Trausan-Matu, Stefan,** Politehnica University of Bucharest, Romania
- **Trzcielinski, Stefan,** Poznan University of Technology, Poland
- **Tsihrintzis, George,** University of Piraeus, Greece
- **Tudoroiu, Nicolae,** John Abbott College, Canada
- **Vega-Rodríguez, Miguel A.,** University of Extremadura, Spain
- **Virvou, Maria,** University of Piraeus, Greece
- **Watanabe, Toyohide,** University of Nagoya

DOI: 10.15439/2021F91
ISSN 2300-5963 ACSIS, Vol. 25

Design and application of facial expression analysis system in empathy ability of children with autism spectrum disorder*

Chen Guo, Kun Zhang**, Jingying Chen, Ruyi Xu, Lei Gao
Faculty of Artificial Intelligence in Education,
National Engineering Research Center for E-Learning,
Central China Normal University, Wuhan, China
Email: zhk@mail.ccnu.edu.cn

Abstract—**Empathy is an important social ability in the early childhood development. One of the significant characteristics of children with autism spectrum disorder (ASD) is their lack of empathy, which makes it difficult for them to feel and understand other people's emotions and to judge other people's behavioral intentions, leading to social disorders. This research designs and implements a facial expression analysis system that could obtain and analyze the real-time facial expressions of children when viewing stimulus materials, and then evaluate the differences of empathy ability between ASD children and typical development (TD) children. The results of this research provide new ideas for the evaluation of ASD children, and also help to develop empathy intervention plans for ASD children.**

I. Introduction

AUTISM Spectrum Disorder (ASD) is a developmental disability that can cause many social, communication and behavioral challenges [1]. The causes of ASD are still unclear, and there is no special medicine for treatment. Only early and long-term educational intervention can improve the ability and behavior of ASD children. It has caused huge difficulties for their study and development, and become a heavy burden on their families and society. In recent years, the evaluation and educational intervention of ASD have received more and more attention [2][3]. However, due to the increasing incidence of ASD worldwide, the number of educational institutions and the effectiveness of educational intervention for individual differences in ASD children are still facing severe challenges.

In the early childhood development, social disorders are regarded as the most obvious symptom of ASD children. Their social disorders could be manifested before the development of language, such as distracted attention and dull facial expressions. Many studies had shown that the lack of empathy was the main factor leading to social disorders in ASD children. The empathy refers to the ability to sense other people's emotions and to imagine what someone else might be thinking or feeling. It is a very important social ability in early childhood development [4]. Due to the lack of empathy, ASD children have difficulty in feeling and understanding the emotions of others and judging the behavioral intentions of others, resulting in social disorders. Therefore, the evaluation research of empathy is helpful to infer the probability of ASD and provide guidance for empathy intervention programs.

Facial expression is one of the main forms in the process of social interaction and can be used as a means of evaluating children's social skills. For typical development (TD) children, they usually have the ability to capture facial expressions when they are about six months old, and can recognize several basic facial expressions when they are about one year old. Some researches had tried to analyze children's facial expressions through quantitative methods. For example, Rozga et al. used physiological sensors to determine the ability of ASD children to imitate other people's expressions [5]. However, when this kind of physiological sensor was in direct contact with ASD children's facial skin, it might inhibit their spontaneous facial expressions [6][7]. Afterwards, Samad proposed to capture facial images of ASD children through optical image sensors, and analyze the movement of ASD children's muscles related to facial expressions when they received visual stimuli, so as to determine their ability to imitate facial expressions [8]. In the process of data collection, the use of non-invasive sensors could eliminate noise interference caused by sensor intrusion, and could more truly reflect the facial expression changes of ASD children.

In recent years, with the emergence of a large number of high-precision facial expression analysis algorithms, some researchers had used facial expression analysis techniques to evaluate the differences in facial expressions of ASD children and TD children [9]. For example, Coco et al. used a computer vision-based facial expression analysis method to quantify the generation of facial expressions, and proved that ASD children and TD children had different facial expressions in response to external stimuli [10].

Based on the promising work proposed by the above studies, this paper proposed a facial expression analysis system to evaluate the empathy ability of children. By collecting facial expression data of ASD children and TD children when they received visual stimuli, the facial expression analysis system was used for preprocessing and

* This work was supported by National Natural Science Foundation of China (No. 61807014) and the Fundamental Research Funds for the Central Universities (No. CCNU20QN026, CCNU19QN039).

** Corresponding author, Email: zhk@mail.ccnu.edu.cn.

facial expression analysis to determine whether ASD children and TD children can produce corresponding emotional responses and the corresponding degree of emotional arousal when receiving external visual stimuli, so as to study the difference in empathy ability between ASD children and TD children.

The second section of this paper presented the system framework and introduced the design of each module. The third section presented the system implementation, including the data processing process. The fourth section introduced the system application, including experimental design, as well as the analysis and discussion of experimental results.

II. System Design

A. System framework

The main function of the facial expression analysis system was to analyze facial expressions in real time through the camera and visualize the analysis results. It was divided into four modules, which were real-time data loading module, data preprocessing module, expression analysis module and results visualization module. The system framework was shown as in Fig. 1.

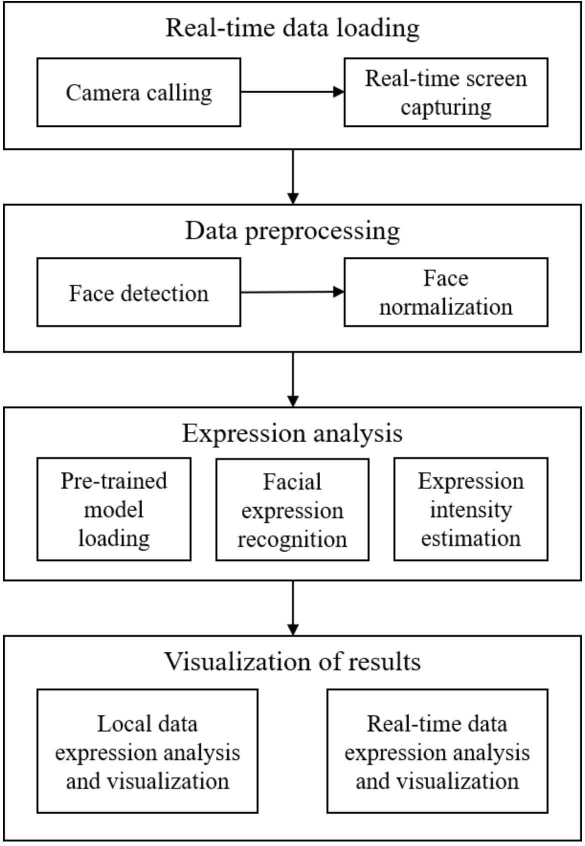

Real-time data loading
Camera calling
Real-time screen capturing
Data preprocessing
Face detection
Face normalization
Expression analysis
Pre-trained model loading
Facial expression recognition
Expression intensity estimation
Visualization of results
Local data expression analysis and visualization
Real-time data expression analysis and visualization

Fig. 1 System framework

B. Real-time data loading module

The real-time data loading module called the currently available camera, read and displayed the frame of the camera, and then input the captured images to the data preprocessing module frame by frame. If there was currently no available camera, this module would pop up a prompt to inform the user. The flowchart of this module was shown as in Fig. 2.

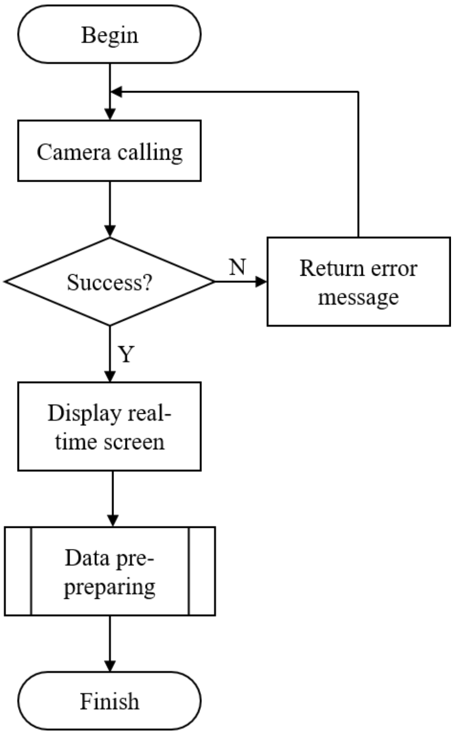


Fig. 2 Real-time data loading flowchart

C. Data preprocessing module

There were usually some noises that were not related to expressions (such as background, clothing, etc.) during the collection of facial expression data. These factors unrelated to expressions would affect the extraction effect of the convolutional neural network on the expression-related features in the images, and affect the expression recognition effect of the model. Therefore, it was necessary to preprocess the expression image before inputting the model. The data preprocessing module of this system included two steps: face detection and face normalization. Face detection was done using MTCNN [11], and face normalization was done using OpenCV image processing tools.

D. Expression analysis module

The expression analysis module had two main functions, namely the loading of the pre-trained model and expression analysis. In order to reduce the waiting time when loading the model, the user could set the storage directory of the model when selecting the function, and the system would complete the loading of the model in advance. The expression analysis function used the model trained in the previous step to perform expression classification and intensity estimation on the input expression samples, and save the results in the corresponding directory. The flowchart of this module was shown as in Fig. 3.

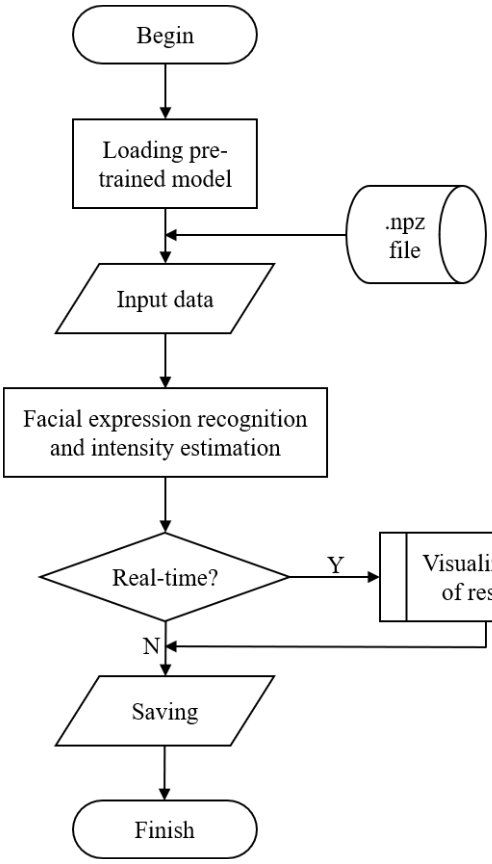

Begin
Loading pre-trained model
.npz file
Input data
Facial expression recognition and intensity estimation
Real-time?
Y
Visualization of results
N
Saving
Finish

Fig. 3 Expression analysis flowchart

E. Results visualization module

After the expression recognition and intensity estimation were completed, the visualized results would be directly displayed in the interface to enhance user experience. The content included the face frame generated by face detection, the results of facial expression recognition and intensity estimation, etc.

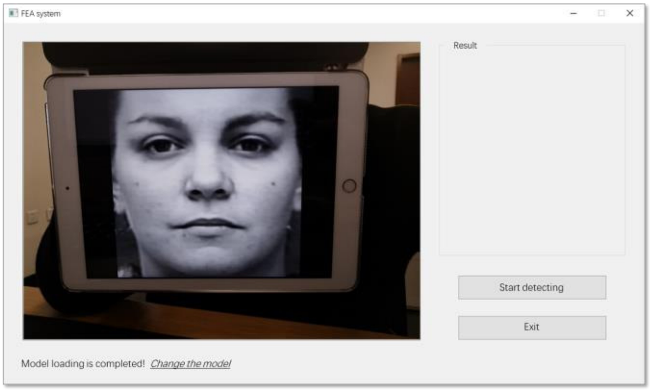
III. System Implementation

A. System development environment

The facial expression analysis system was developed on the 64-bit Windows 10 version 1803, using PyCharm 2019 development environment, Python version 3.6.6 and PyQt5 toolkit for coding. Finally, the PyInstaller toolkit was used to package the code files to generate executable files. The hardware environment was Intel Core i7-8700K CPU, with 64 GB memory and 2 TB disk space.

B. Real-time data display

When the user selected the real-time detection function on the main window, the system entered the real-time data display interface. The Video Capture method in the OpenCV library was used to call the camera and display real-time images. If no cameras were available, a prompt would pop up to tell the user. When the system could capture the camera image normally, the user could click the start detection button to start the real-time recognition function. and the system would input the captured image frame by frame into the preprocessing module for image processing. The real-time display interface was shown as in Fig. 4.



Model loading is completed! *Change the model*

Fig. 4 Real-time data display interface

C. Data preprocessing

The data preprocessing was mainly divided into two steps. First, the face detection algorithm was used to detect the facial feature points and the face marker frames. Second, the face normalization was done using the detection results, to obtain the standard facial expression image. The specific methods of face detection and normalization were as follows:

(1) Face detection

The cascaded multi-task convolutional neural network (MTCNN) was used for face detection. The algorithm was carried out in three steps. First, the Proposal-Net was used to quickly generate face location candidate frames. Second, the Refine-Net was used to optimize the facial candidate frames generated in the first step, eliminate the incorrect frames, and capture the high-quality candidate frames through the NMS algorithm. Third, the Output-Net was used to locate and mark five key feature points of the face while removing overlapping candidate frames. Face detection was performed on the data through MTCNN algorithm, and a series of labeled data with face detection results and 5 key feature points were obtained.

(2) Face normalization

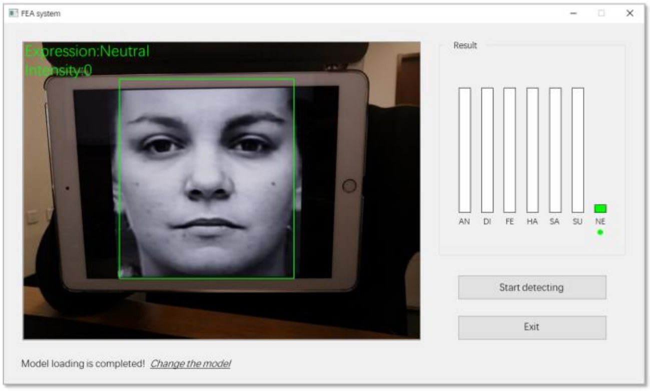
According to a series of face coordinates and feature point information obtained by face detection, most of the noises that were not related to facial expressions in the image could be eliminated after cropping. Because the distance between the participant's face and the collection device could not be fixed during data collection, the size of the face extracted in the face detection step was different. But the deep network model required input images of uniform size. Therefore, it was necessary to normalize the image to obtain an expression image of a uniform size. In addition, in order to retain the areas that were highly related to facial expressions such as the corners of the mouth and the eyes after image cropping and size normalization, the image needed to be geometrically preprocessed so that the processed image could maintain the same proportions as the original image. The specific method was to extract the coordinate points of the left eye and the right eye and the coordinate data of the face detection frame from the feature point mark data, call the fitgeotrans function in OpenCV to perform geometric transformation and fitting of the coordinate point pair group, align and crop the image using the normalization parameters and the coordinates of the face detection frame. Finally, the face image was normalized to 128×128 pixels.

D. Expression analysis

When the real-time image of the camera was displayed, the user could click the start detection button to input the image sequence into the expression analysis module for expression recognition and intensity estimation using the loaded model, and output the recognition result to the visualization module and save to the specified path. If the system did not detect a face in the captured image, it would output "No face detected" on the screen to prompt the user.

E. Results visualization

When the real-time detection was working, if the facial expression was successfully detected by the analysis module, the result would be displayed according to the intensity and category of the expression. The face frame for face detection was displayed on the real-time screen, and the expression category and intensity were displayed in the upper left corner of the real-time screen. On the right side of the screen, the face detection results were displayed in the form of a histogram and marked under the corresponding category. The figure was drawn using the matplotlib package in the Python toolkit. The horizontal axis represented the Neutral Expression (NE) and six types of basic expressions, which were Anger (AN), Disgust (DI), Fear (FE), Happiness (HA), Sadness (SA), Surprise (SU). The vertical axis represented the intensity of these six types of basic expressions. When the expression intensity was zero, that indicated a neutral expression, the sample was displayed in the NE column by default. The results visualization interface was shown as in Fig. 5.



Expression:Neutral
Intensity:0

Model loading is completed! Change the model

Fig. 5 Results visualization interface

IV. System Application

The facial expression analysis system proposed in this paper was used to evaluate the empathy ability of children. Firstly, dynamic videos of facial expressions of ASD children and TD children when receiving visual stimuli were collected, intercepted and saved in frames, and organized into a data set of children's expressions. Then the facial expression analysis system was used to analyze the children's expression data set to determine whether ASD children and TD children could understand and produce corresponding emotional responses when receiving external visual stimuli, and the degree of corresponding emotional arousal. These would serve as the basis for evaluating children's empathy ability.

A. Experimental materials

Many studies had shown that the facial expression disorder of ASD children was usually manifested as the expression recognition disorder to negative expressions (such as anger, fear, etc.) was often greater than that of positive expressions (such as happiness, surprise, etc.). Liping Gu proposed that children's ability to recognize expression was usually affected by the emotional intensity of pictures, leading to differences in the results of children's recognition of expressions in pictures [12]. Generally, pictures with positive expressions and high emotional intensity were more likely to be recognized, while children had lower ability to recognize negative expression and pictures with low intensity [13]. Therefore, in order to make the collected data have a more obvious distinction, five segments of positive emotion stimulus materials were selected from the initial material library for the experiment. These materials were all from the animated version of the children's themed sitcom, called "Family with Children". Each video segment was selected by experts in the special education field after analyzing and evaluating factors such as video duration, expression type, difficulty in understanding, and emotional arousal. Each selected segment was about 12 seconds long, and the video themes included: being praised, helping others, rehearsing the program, going home from school, and eating snacks.

B. Experimental participates

The children who participated in the data collection of the experiment were aged 3-7 and came from a kindergarten (30 children in TD group) and a special children's rehabilitation institution (30 children in ASD group) in Wuhan. All children in ASD group were clinically diagnosed with ASD (according to DSM-5, 2013), normal or corrected visual acuity, and no other respiratory diseases, childhood schizophrenia, epilepsy and other organic brain diseases.

Before the experiment, the agreements were signed with the kindergarten and institution, and the informed consent forms were signed with their parents to protect the privacy of these children participating in the experiment.

C. Experimental design

In order to avoid the mutual interference of different video materials on emotional awakening, each video segment was played after an interval of 30 seconds from the end of the previous segment. A question session would be conducted on the content of the video material to determine whether the child understood the material in the video.

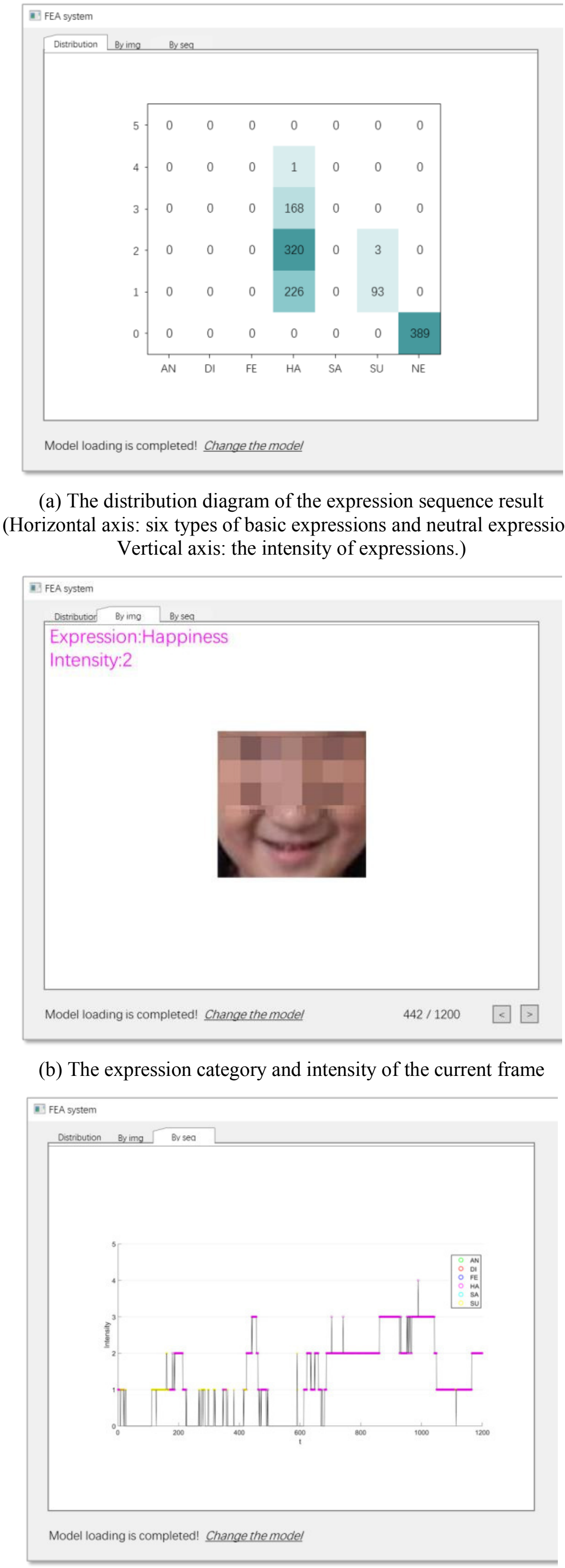
When children watched the stimulus, the camera recorded their emotional response. A total of 60 segments of effective video data were collected from 30 ASD children and 30 TD children. The video resolution was 720×576 pixels. The video processing module in OpenCV was used to process the video data and save it as the image sequence by frame. Each image sequence contained about 1200 frames, which recorded the spontaneous changes in expression of each child under same emotional stimulus. The psychologists labeled each sequence with the child's main facial expression category.

In this experiment, a pre-trained model was used to test all frames in the children's data set. The model was trained on the CK+ dataset using the expression analysis method of the tensorflow-gpu 1.13.1 framework. The number of iterations was set to 3000 poaches. The early stopping method was used in the training process to ensure that the model got the best generalization ability and could better fit the data. When the accuracy of the validation set dropped, the training process was terminated and the model was saved for use in the children's facial expression detection.

D. Experimental results and analysis

The facial expression analysis system was used to analyze the collected children's expression data set to explore the difference between the empathy ability of ASD children and TD children. Fig. 6 showed the experimental results of a certain child's expression sequence analyzed by the facial expression analysis system, where Fig. 6(a) was the result of the distribution diagram of the expression sequence, Fig. 6(b) was the expression category and intensity of the current frame, Fig. 6(c) was the expression category and intensity result of the sequence.

It could be seen that in Fig. 6(a), the overall distribution of the data was statistics. Among them, the accuracy rate referred to the percentage of the children's facial expression data label consistent with the expression category predicted by the model in the overall sample. The tracking rate referred to the ratio of the emotional feedback generated by the child on the facial expression which was consistent with the emotion of the stimulus material. The child shown in the picture could better understand the emotion expressed by the stimulus material and could reflect it on facial expressions. Therefore, it could be considered that this child had a good empathy ability. Fig. 6(b) could help users analyze the wrong samples by viewing each frame of image sequence. Fig. 6(c) showed the overall recognition of the child's facial expression sequence. This function could be used to analyze whether the child had produced the corresponding emotion under the corresponding stimulus material in combination with the video material.



	AN	DI	FE	HA	SA	SU	NE
5	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0
3	0	0	0	168	0	0	0
2	0	0	0	320	0	3	0
1	0	0	0	226	0	93	0
0	0	0	0	0	0	0	389

Model loading is completed! *Change the model*

(a) The distribution diagram of the expression sequence result (Horizontal axis: six types of basic expressions and neutral expression. Vertical axis: the intensity of expressions.)

Expression:Happiness
Intensity:2

Model loading is completed! *Change the model* 442 / 1200

(b) The expression category and intensity of the current frame

Model loading is completed! *Change the model*

(c) The expression category and intensity result of the sequence (Horizontal axis: frame sequence number. Vertical axis: the intensity of expressions.)

Fig. 6 Experimental results of a certain child's expression sequence

TABLE I.
ANALYSIS RESULTS OF ALL CHILDREN'S FACIAL EXPRESSIONS

Group	Accuracy rate	Tracking rate
Autism Spectrum Disorder (ASD)	90.0%	23.3%
Typical Development (TD)	93.3%	80.0%

The results of all children's expression data analyzed by the facial expression analysis system were shown in Table I. It could be concluded that these two groups of children could produce corresponding feedback when they receive the same emotional stimulus materials, but the proportion of children in the TD group that produced the same emotion as the stimulus was significantly higher than that in the ASD group. It could be inferred that there were significant differences in emotional understanding and facial expression recognition between TD children and ASD children.

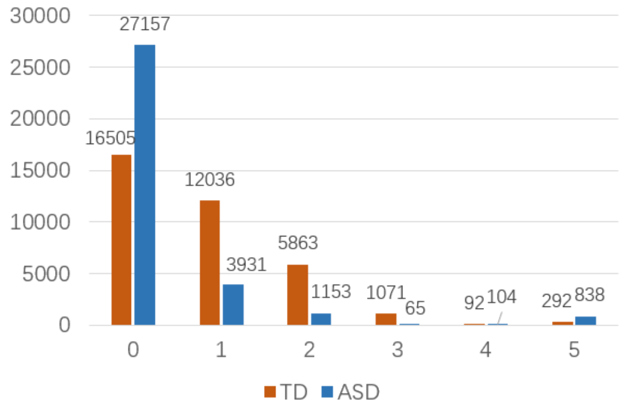


Fig. 7 The statistics of the expression intensity results (Horizontal axis: the intensity of expressions. Vertical axis: the number of samples.)

Fig. 7 was the statistics of the expression intensity results of the TD group and the ASD group. The expression intensity was divided into 6 levels (from 0 to 5). It could be found that in samples with an intensity of 0, the number of samples in ASD group was significantly more than that in TD group. For the samples with expression intensity in the interval [1,3], the number of samples in TD group was significantly more than that in ASD group. It showed that the same stimulus material had higher emotional arousal to TD children than to ASD children, and most ASD children could not produce correct feedback when receiving external emotional stimuli. From the perspective of empathy, there might be two reasons. On the one hand, ASD children might have defects in feeling and experiencing the emotions of others, which made it difficult or even impossible for them to understand other people's emotions. On the other hand, ASD children might have defects in expressing emotions and could not express their true inner emotions through facial expressions. In the interval between expression intensity 4 and 5, the number of samples in ASD group was slightly higher than that in TD group. This might be because many ASD children were often accompanied by symptoms such as distracted attention and impulsiveness. When viewing stimulus materials, there might be interference factors such as large changes in head posture or occlusion of relevant facial expressions in ASD children, leading to errors in the expression intensity identification of some samples.

In order to evaluate the empathy ability of the two groups of children more comprehensively, the temporal statistical method of expression category and expression intensity was used for further analysis. Specifically, the detailed criteria for evaluating empathy (E) were set as follows: (1) According to the research work in Reference [14], whether child's main facial expression category (C) was consistent with emotional stimuli (S) was determined by analyzing the consistency of the child's facial expression and the emotion of the stimulus. (2) Inspired by Reference [15], the expression awakening duration was used as the time feature to determine whether the expression activation duration (A) exceeded the threshold. In this experiment, the threshold was set to 1/3. When the total duration of samples whose facial expression intensity was greater than 0 exceeded the threshold, the expression was considered to have been activated.

$$\mathrm{E} = \begin{cases} 2 & A > \frac{1}{3}, C = S \\ 1 & A > \frac{1}{3}, C \neq S \mid A \leq \frac{1}{3}, C = S \\ 0 & A \leq \frac{1}{3}, C \neq S \end{cases} \tag{1}$$

For empathy indicators, 0, 1, and 2 respectively represent the low, medium, and high levels of empathy ability. The larger the value, the stronger the empathy ability. According to the above criteria, the children in the ASD group and the TD group could be classified and counted. The results were shown in Table II.

TABLE II.
EVALUATION RESULTS OF CHILDREN'S EMPATHY ABILITY

Expression category consistency	Activation duration	TD group	ASD group	Empathy Level
C=S	A>1/3	22	0	2
	A≤1/3	2	7	1
C≠S	A>1/3	6	3	1
	A≤1/3	0	20	0

It could be seen that in the TD group, 22 children were evaluated as high-level empathy ability, and 8 children were evaluated as medium-level. In the ASD group, 20 children had the low-level results of empathy ability, and 10 children had the medium-level results. The experimental results further proved the effectiveness of this expression analysis system in evaluating children's empathy ability.

V. CONCLUSION

This paper introduced the design and implementation of the facial expression analysis system, which could quickly recognize and estimate the intensity of facial expression data captured by the camera, provide visualization of the results, providing a great convenience for users to analyze facial expression. This system was used to evaluate children's empathy ability and had achieved good results. It verified the effectiveness of the proposed method and the application value of this system. The results of this research provided a certain basis for the evaluation of children's empathy ability, and also helped to develop empathy intervention programs for ASD children.

Future research work is to further explore and improve the application of facial expression analysis in the evaluation of children's empathy ability, such as expanding the scale of experimental data to obtain more applicable conclusions, and optimizing system algorithms to improve recognition accuracy. It is also possible to combine facial expression analysis with EEG signal and other methods to evaluate the children's empathy ability in multiple dimensions, so as to obtain more accurate evaluation results.

REFERENCES

[1] American Psychiatric Association, "Diagnostic and Statistical Manual of Mental Disorders: DSM-V," Washington, DC: American Psychiatric Publishing, pp. 55-59, 2013. ISBN: 978-0-89042-554-1.

[2] K. Pancerz, W. Paja and J. Gomuła, "Random forest feature selection for data coming from evaluation sheets of subjects with ASDs," 2016 Federated Conference on Computer Science and Information Systems (FedCSIS), 2016, pp. 299-302. http://dx.doi.org/10.15439/2016F274.

[3] A. Kołakowska, A. Landowska, M. R. Wrobel, et al, "Applications for investigating therapy progress of autistic children," 2016 Federated Conference on Computer Science and Information Systems (FedCSIS), 2016, pp. 1693-1697. http://dx.doi.org/10.15439/2016F507.

[4] Xiaoxia Zhang, Ye Wang, Xin Liu, et al, "A Review of Studies on Empathy Development of People with Autistic Spectrum Disorders," Chinese Journal of Special Education, vol. 8, pp. 48-55, 2019. (in Chinese). http://dx.doi.org/10.3969/j.issn.1007-3728.2019.08.009.

[5] Rozga A, King T Z, Vuduc R W, et al, "Undifferentiated facial electromyography responses to dynamic, audio-visual emotion displays in individuals with autism spectrum disorders," Developmental Science, vol. 16(4), pp. 499-514, 2013. http://dx.doi.org/10.1111/desc.12062.

[6] Beall P M, Moody E J, Mcintosh D N, et al, "Rapid facial reactions to emotional facial expressions in typically developing children and children with autism spectrum disorder," Journal of Experimental Child Psychology, vol. 101(3), pp. 206-223, 2008. http://dx.doi.org/10.1016/j.jecp.2008.04.004.

[7] Mathersul D, Mcdonald S, Rushby J A, "Automatic facial responses to affective stimuli in high-functioning adults with autism spectrum disorder," Physiology & Behavior, vol. 109(1), pp.14-22, 2013. http://dx.doi.org/10.1016/j.physbeh.2012.10.008.

[8] Samad M D, Bobzien J L, Harrington J W, et al, "Non-intrusive optical imaging of face to probe physiological traits in Autism Spectrum Disorder," Optics & Laser Technology, vol. 77, pp. 221-228, 2016. http://dx.doi.org/10.1016/j.optlastec.2015.09.030.

[9] Leo M, Carcagnì P, Distante C, et al, "Computational assessment of facial expression production in ASD children," Sensors, vol. 18(11), pp. 3993, 2018. http://dx.doi.org/10.3390/s18113993.

[10] Coco M D, Leo M, Carcagni P, et al, "A Computer Vision Based Approach for Understanding Emotional Involvements in Children with Autism Spectrum Disorders," International Conference on Computer Vision Workshop, pp. 1401-1407, 2017. http://dx.doi.org/10.1109/ ICCVW.2017.166.

[11] Zhang K, Zhang Z, Li Z, et al, "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks," IEEE Signal Processing Letters, vol. 23(10), pp. 1499-1503, 2016. http://dx.doi.org/ 10.1109/LSP.2016.2603342.

[12] Liping Gu, Jin Jing, Yu Jin, et al, "Research of the relationship between the ability of emotion understanding and social adaptation in children with high functioning autism," Chinese Journal of Child Health Care, vol. 21(1), pp. 16-19, 2013. (in Chinese)

[13] Hou Hu, Mingfan Wu, Shenghua Hu, "Facial emotion recognition in children with high functioning autism," Chinese Journal of School Health, vol. 35(8), pp. 1146-1149, 2014. (in Chinese)

[14] Anastassiou-Hadjicharalambous X, Warden D, "Convergence between physiological, facial and verbal self-report measures of affective empathy in children," Infant and Child Development: An International Journal of Research and Practice, vol. 16(3), pp. 237-254, 2007. http://dx.doi.org/10.1002/icd.464.

[15] Li B, Mehta S, Aneja D, et al, "A Facial Affect Analysis System for Autism Spectrum Disorder," IEEE International Conference on Image Processing, pp. 4549-4553, 2019. http://dx.doi.org/10.1109/ICIP.2019. 8803604.

Advances in Network Systems and Applications

THE rapid development of computer networks including wired and wireless networks observed today is very evolving, dynamic, and multidimensional. On the one hand, network technologies are used in virtually several areas that make human life easier and more comfortable. On the other hand, the rapid need for network deployment brings new challenges in network management and network design, which are reflected in hardware, software, services, and security-related problems. Every day, a new solution in the field of technology and applications of computer networks is released. The ANSA technical session is devoted to emphasizing up-to-date topics in networking systems and technologies by covering problems and challenges related to the intensive multidimensional network developments. This session covers not only the technological side but also the societal and social impacts of network developments. The session is inclusive and spans a wide spectrum of networking-related topics.

The ANSA technical session is a great place to exchange ideas, conduct discussions, introduce new ideas and integrate scientists, practitioners, and scientific communities working in networking research themes.

TOPICS

- Networks architecture
- Networks management
- Quality-of-Service enhancement
- Performance modeling and analysis
- Fault-tolerant challenges and solutions
- 5G developments and applications
- Traffic identification and classification
- Switching and routing technologies
- Protocols design and implementation
- Wireless sensor networks
- Future Internet architectures
- Networked operating systems
- Industrial networks deployment
- Software-defined networks
- Self-organizing and self-healing networks
- Mulimedia in Computer Networks
- Communication quality and reliability
- Emerging aspects of networking systems

TRACK CHAIRS

- **Armando, Alessandro,** University of Genova, Italy
- **Awad, Ali Ismail,** Luleå University of Technology, Sweden
- **Furtak, Janusz,** Military University of Technology, Poland
- **Suri, Niranjan,** Institute of Human and Machine Cognition, United States

PROGRAM CHAIRS

- **Awad, Ali Ismail,** Luleå University of Technology, Sweden
- **Furtak, Janusz,** Military University of Technology
- **Hodoň, Michal,** University of Žilina, Slovakia

PROGRAM COMMITTEE

- **Ahad, M**ohd Abdul, Department of Computer Science and Engineering, Jamia Hamdard, New Delhi
- **Ajlouni, Naim,** Istanbul Aydin University, Turkey
- **Antkiewicz, Ryszard,** Military University of Technology, Poland
- **Brida, Peter,** University of Zilina, Slovakia
- **Bridova, Ivana,** University of Zilina, Slovakia
- **Brzoza-Woch, Ada,** AGH University of Science and Technology, Poland
- **Chaganti, Raj,** ExpediaGroup Inc, Seattle, USA
- **Chmielewski, Mariusz,** National Cyber Security Centre, Poland
- **Chumachenko, Igor,** Kharkiv National University of Municipal Economy named after Beketov, Ukraine
- **Cui, Huanqing,** Shandong University of Science and Technology, China
- **Davidsson, Paul,** Malmö University, Sweden
- **Dotsenko, Sergii,** Ukrainian State University of Railway Transport, Ukraine
- **Długosz, Rafał,** UTP University of Science and Technology, Poland
- **Elmougy, Samir,** Mansoura University, Egypt
- **Faria, Lincoln,** Department of Computer Science, Fluminense Federal University, Brazil
- **Farooq, Ali,** University of Turku, Finland
- **Fouchal, Hacene,** University of Reims Champagne-Ardenne, France
- **Gheisari, ,** Mehdi, Southern University of Science and Technology, China
- **Karpiš, Ondrej,** University of Žilina, Slovakia
- **Kochláň, Michal,** University of Žilina, Slovakia
- **Lavrov, Eugeniy,** Sumy State University, Ukraine
- **Molenda, Karol,** National Cyber Security Centre, Poland
- **Monov, Vladimir V.,** Bulgarian Academy of Sciences, Bulgaria
- **Murawski, Krzysztof,** Military University of Technology, Poland
- **Niewiadomska-Szynkiewicz, Ewa,** Research and Academic Computer Network (NASK), Institute of Control and Computation, Poland
- **Papaj, Jan,** Technical university of Košice, Slovakia
- **Salem, Abdel-Badeeh M.,** Ain Shams University, Egypt

- **Sudhir Kumar Sharma,** Guru Gobind Singh Indraprastha University, New Delhi, India
- **Smolarz, Andrzej,** Lublin University of Technology, Poland
- **Grigore Stamatescu,** University “Politehnica” of Bucharest , Romania
- **Sergiy Tymchuk,** Kharkiv National Technical University of Agriculture , Ukraine
- **Konrad Wrona,** NATO Communications and Information Agency , Poland
- **Zbigniew Zieliński,** Military University of Technology, Poland

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 329–335

DOI: 10.15439/2021F65
ISSN 2300-5963 ACSIS, Vol. 25

Discovering Communities in Networks: A Linear Programming Approach Using Max-Min Modularity

Arman Ferdowsi
Vienna University of Technology
Institute of Computer Engineering
Embedded Computing Systems
Email: aferdowsi@ecs.tuwien.ac.at

Alireza Khanteymoori
University of Freiburg
Department of Computer Science
Bioinformatics Group
Email: alireza@informatik.uni-freiburg.de

***Abstract*—Community detection is a fundamental challenge in network science and graph theory that aims to reveal nodes' structures. While most methods consider Modularity as a community quality measure, Max-Min Modularity improves the accuracy of the measure by penalizing the Modularity quantity when unrelated nodes are in the same community. In this paper, we propose a community detection approach based on linear programming using Max-Min Modularity. The experimental results show that our algorithm has a better performance than the previously known algorithms on some well-known instances.**

I. Introduction

IN MANY (complex) networks, there are sets of nodes with some common characteristics. More specifically, there are sets of highly interactive vertices that are likely to yield and share common relationships and properties among themselves. These sets are called *communities*. Detecting communities has become one of the fundamental subjects in the field of network science and graph theory and has numerous applications in a wide range of areas, including the analysis of Social Network [1], [2], Biological Networks [3], Cosmological Networks [4], and WEB [5]. It also plays a crucial role in the domain of Signal Processing [6], Image Segmentation [7], Pattern Recognition [8], and Data Clustering [9].

A network is basically given as a graph $G = (V, E)$ with the set of vertices V and edges E. A *community* in the network can then be contemplated as a subset of vertices $C \subseteq V$ with a high density of edges between nodes inside the subset and a low density of edges connecting this subset to the others. Accordingly, one can define the community detection problem as *partitioning* V into a set of disjoint communities $\mathbf{C} = \{C_1, C_2, \dots, C_k\}$. In the literature, several quality measures can be used to qualify the goodness of a partitioning. One of the most widely used and well-known quality measures is *Modularity*, introduced by Newman [10]: Let $A = (a_{i,j})$ be the adjacency matrix of G, where $a_{i,j}$ is one when there is an edge between node i and node j, and zero otherwise; $d_i = \Sigma_{l=1}^{n} a_{i,l}$ be the degree of node i; m be the number of edges and n be the number of vertices in G. Modularity Q of a given partitioning $\mathbf{C}$ is defined as:

$$Q(\mathbf{C}) = \frac{1}{2m} \sum_{i,j \in V} [a_{i,j} - \frac{d_i d_j}{2m}] \sigma(i,j) \tag{1}$$

where $\sigma(i,j)$ is one if i and j are in the same community and zero otherwise.

Intuitively, for a community C, Modularity is the number of edges within C minus the expected number of such edges. So, the high-quality communities can be determined as the ones with the high value of Modularity. We refer to the problem of finding a partition of the network that maximizes Modularity as the *Modularity Maximization* problem. The Modularity Maximization problem is NP-hard [11]. Nevertheless, many algorithms, both heuristics (e.g., [12], [13], [14], [15], [16]) and exact methods (e.g., [11], [17], [18]) have been proposed to solve this problem (approximately).

It is known that the Modularity measure suffers from some limitations (see [19], [20] for more details). In particular, as pointed out in [21] and [22], one of the major limitations of Modularity is that it only takes the existing edges of the network into consideration. In other words, Modularity qualifies the goodness of the discovered communities by only measuring how good the partitioning fits the existing edges. This is indeed a drawback because Modularity does not consider the disconnected nodes (absent edges) that lie in the same community. *Max-Min Modularity* [21] is one of the successful extensions of Modularity which improves the accuracy of the measure by penalizing the Modularity quantity when disconnected nodes are in the same community. More precisely, it is assumed in [21] that (in addition to the graph G) a zero-one *relation matrix* $U = (u_{i,j})$ is given that defines whether every pair of disconnected nodes of the network is related or not; where $u_{i,j}$ is one when disconnected nodes i and j are *related*, and zero otherwise. They, in fact, take into account the importance of the *indirect* connections between disconnected nodes by only penalizing the Modularity measure when *unrelated* nodes are in the same community: Consider a complemented graph $G' = (V, E')$, where E' contains an edge between every pair of disconnected nodes of G that is unrelated; i.e., there is an edge between i and j in G' if there is not such an edge in G and also $u_{i,j}$ is zero. Let $A' = (a'_{i,j})$ be the adjacency matrix of G' and d'_i be the degree of node i in G' accordingly. Let m' be the number of the edges in G'. Max-Min Modularity Q_{MM} of a given partition $\mathbf{C}$ of V is defined as follows:

$$Q_{MM}(\mathbf{C}) = \sum_{i,j \in V} [\frac{1}{2m}(a_{i,j} - \frac{d_i d_j}{2m}) - \frac{1}{2m'}(a^{'}_{i,j} - \frac{d^{'}_i d^{'}_j}{2m'})]\sigma(i,j) \quad (2)$$

We refer to the problem of finding a partition of the network that maximizes Max-Min Modularity as the *Max-Min Modularity Maximization* problem. Chen et al. [21] proposed a hierarchical clustering algorithm (similar to that of Newman [10] for the classical Modularity Maximization Problem) that approximately optimizes Max-Min Modularity in a greedy manner.

A drawback of the approach described in [21] is that it strongly depends on the accuracy of the given relation matrix. So the quantity of the measure might be heavily affected by the node relationships defined by the user in the first place. Therefore, unobserved or misobserved relations between nodes of the network can lead to poor partitioning results. It is worth mentioning that authors of [21] also suggested a systematic (but not necessarily accurate) way for defining the relation matrix U: Two disconnected nodes are related if they connect to the same intermediary node; this is, for every node pairs i and j, $u_{i,j}$ is one only if $\{i, j\} \notin E$ and there is some node k that $\{i, k\} \in E$ and $\{k, j\} \in E$, and zero otherwise.

Main contribution: We develop the first LP-based approach for solving the Max-Min Modularity Maximization problem. First, we provide a more accurate way of defining the relation matrix by exploiting an optimal linear relaxation solution to the standard integer linear programming of the Modularity Maximization problem. After that, we depict the standard integer programming formulation of the Max-Min Modularity Maximization problem. Then, for solving the problem, we employ a row and column generation approach to efficiently solve the linear programming relaxation the problem. This provides an optimal fractional solution to the Max-Min Modularity Maximization problem. Next, we design a new rounding algorithm to obtain integer solutions and, therefore, to determine the community structures. We finally present a computational study of our algorithm on known instances. The computational experiments show that our results highly resemble the optimal solutions and that our algorithm outperforms the previous well-known algorithms, including the algorithm proposed in [21].

The paper is organized as follows: the rest of this section focuses on providing a brief literature review. In Section II, we first introduce the novel relation matrix, and then we model the Max-Min Modularity Maximization problem based on that. Next, in Section III, we depict the row/column generation technique and also the local search-based rounding algorithm. Section IV is then dedicated to the experimental results.

A. Related Works

In the literature, several approaches are proposed to detect communities in the networks: extremal optimization [23], spectral optimization [24], greedy heuristics [25], [26], simulated annealing [27], dynamical clustering [28], deep learning techniques [29], message passing [30], quantum mechanics [31], and more.

Despite a considerable amount of work on the community detection problem, relatively little work solves the problem using linear programming or integer programming techniques. In 2008, Agarwal and Kempe [32] expressed the Modularity Maximization problem as a standard Integer Programming (IP) model and proposed an LP rounding algorithm for the problem. Although the LP relaxation of their model can be solved in polynomial time, as the number of constraints in their model is $O(n^3)$, the rounding algorithm becomes impractical when the number of nodes is large. Consequently, in 2010, a column generation technique is developed in [17] to solve the model more efficiently . Nevertheless, the proposed algorithm could not solve problems with more than a few hundred nodes in a reasonable time. In 2011, Dinh and Thai [33] proposed a sparse LP formulation for the problem with much fewer constraints than that of [32]. Finally, in 2013, Miyamoto [34] proposed a row and column generation approach to solve the sparse LP formulation, resulting in an efficient algorithm for obtaining the optimal value of the sparse LP relaxation (and so an upper bound for the optimal value for the Modularity Maximization problem).

II. Model Description

Let the binary variable x_{ij} indicate if nodes i and j belong to the same community or not; the value of x_{ij} is zero if nodes i and j belong to the same community, and one otherwise. Let $I_{all} = \{(i, j) \in V^2 \mid i < j\}$; and $q_{ij} = a_{i,j} - \frac{d_i d_j}{2m}$, for each $(i, j) \in I_{all}$. As described in [33], the Modularity Maximization problem can be formulated in terms of the following integer linear program.

$$\begin{aligned} \max \quad & \frac{1}{m} \sum_{(i,j)\in I_{all}} q_{ij}(1 - x_{ij}) && && \text{(IP-M)} \\ & x_{ij} + x_{jk} - x_{ik} \geq 0 && \forall i < j < k && (3) \\ & x_{ij} - x_{jk} + x_{ik} \geq 0 && \forall i < j < k && (4) \\ & -x_{ij} + x_{jk} + x_{ik} \geq 0 && \forall i < j < k && (5) \\ & x_{ij} \in \{0, 1\} && \forall (i, j) \in I_{all} && (6) \end{aligned}$$

Constraints (3)-(5) guarantee that if i and j are in the same community and j and k are in the same community, then so are i and k. We refer to the relaxation of (IP-M), obtained by replacing the constraints $x_{ij} \in \{0, 1\}$ by $x_{ij} \in [0, 1]$, as (LP-M).

A. Computing the Relation Matrix via LP

In this section, we provide a systematic and accurate way for defining the relation matrix by exploiting an optimal solution to (LP-M). Let x^* be the optimal solution to (LP-M). This can be obtained efficiently (in polynomial time) using, for example, the row and column generation algorithm of [34]. We note that the optimal fractional solution x^* induces a metric, called *the LP distance*, on the graph G: think of x^*_{ij} as a "*distance*" between nodes i and j. Observe that Constraints

(3)-(5) guarantee the *triangle inequality* for any $i, j, k \in V$ in the induced metric. Clearly, the larger the LP distance of two nodes is, the less related the nodes are. This observation and also the fact that the Modularity Maximization problem can be nicely proposed for weighted graphs [35] motivates us to define the relation matrix and so the complemented (weighted) graph G' using the LP distance (rather than the graph distance). Recall that Chen et al. [21] defined the relation matrix using the distance between nodes in the graph G: Two disconnected nodes are related if they connect to the same intermediary node (if the distance between them in G is two).

We define the relation matrix matrix $A' = (a'_{i,j})$ (and hence G'; $(a'_{i,j})$ represents the weight of the edge between nodes i and j in G') as follows:

$$a'_{i,j} = \begin{cases} x^*_{ij} & \text{if } a_{i,j} = 0 \text{ and } j > i \\ x^*_{ji} & \text{if } a_{i,j} = 0 \text{ and } i > j \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Before any further discussion, it is worth pointing out that only by replacing our proposed relation matrix with the one used in [21] and then applying their hierarchical algorithm we can attain more accurate results. Fig. 1 proves this claim in the following way. It considers 12 well-known networks whose optimal communities (*ground truth*) are already known and valid. It then provides a comparison between each network's ground truth and the communities discovered by the Max-Min Modularity method with respect to $i)$ the conventional relation matrix U, proposed in [21], (red diagram), and $ii)$ our proposed relation matrix (gray diagram). Section IV describes the networks used and the performance metric *Normalized Mutual Information (NMI)*. However, for now, note that for a given network with known community assignments and a given community detection algorithm, the more the NMI value, which can vary between 0 and 1, the more similarity there is between the discovered communities and the ground truth. The results clearly illustrate that applying the proposed relation matrix leads to more accurate communities, which are more similar to the ground truth.

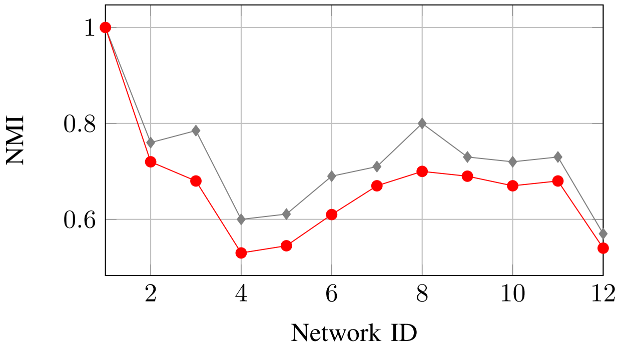


Fig. 1: Comparison between NMI values, for twelve well-known real-world networks, achieved by (i) red curve: Max-Min modularity, proposed in [21] and (ii) gray curve: using our proposed relation matrix but the hierarchical algorithm proposed in [21].

Considerably attractive is that one can still significantly improve the results by solving the standard formulation of the Max-Min Modularity Maximization problem, which will be explained in the following sub-section.

B. Modeling the Max-Min Modularity Maximization problem

First of all, note that it is not difficult to check that the redundant constraints introduced in [36] for the clique partitioning problem are also redundant for the standard formulation of the Modularity Maximization problem and the problem of Max-Min Modularity Maximization. Accordingly, for a given matrix $A' = (a'_{i,j})$, resp. the weighted graph G', defined above, the Max-Min Modularity Maximization problem can be formulated as the following IP. Let $c_{ij} = \frac{q_{ij}}{m} - \frac{q'_{ij}}{m'}$, where $q'_{ij} = a'_{i,j} - \frac{d'_i d'_j}{2m'}$, $d'_i = \sum_{l=1}^{n} a'_{i,l}$, and $m' = \sum_{(i,j)\in I_{all}} a'_{i,j}$; for each $(i, j) \in I_{all}$.

$$\max \sum_{(i,j)\in I_{all}} c_{ij}(1 - x_{ij}) \tag{IP-MM}$$

$$x_{ij} + x_{jk} - x_{ik} \geq 0 \qquad \forall i < j < k,\ c_{ij} \geq 0 \vee c_{jk} \geq 0 \tag{8}$$

$$x_{ij} - x_{jk} + x_{ik} \geq 0 \qquad \forall i < j < k,\ c_{ij} \geq 0 \vee c_{ik} \geq 0 \tag{9}$$

$$-x_{ij} + x_{jk} + x_{ik} \geq 0 \qquad \forall i < j < k,\ c_{jk} \geq 0 \vee c_{ik} \geq 0 \tag{10}$$

$$x_{ij} \in \{0, 1\} \qquad \forall (i, j) \in I_{all} \tag{11}$$

We refer to the relaxation of (IP-MM), obtained by replacing the constraints $x_{ij} \in \{0, 1\}$ by $x_{ij} \in [0, 1]$, as (LP-MM).

III. Solution Approach

To solve (IP-MM), we first employ a technique to find the optimal solution to (LP-MM) efficiently. Then we propose a local search-based rounding procedure to obtaining the integer solution.

A. Solving (*LP-MM*)

While (LP-MM) can be solved in polynomial time, it would be deficient for networks exceeding a few hundred vertices since the number of constraints is $3\binom{n}{3} = O(n^3)$, and therefore, rapidly grows with respect to the number of nodes. To tackle this difficulty, we introduce a row/column generation technique heavily inspired by the one proposed in [34] for the Modularity Maximization problem. Let $I = \{(i, j) \in I_{all} \mid c_{ij} > 0\}$ and $I' = \{(i, j) \in I_{all} \mid c_{ij} \leq 0\}$ be two sets of vertex pairs indices in I_{all}. For a given $\mathcal{I} \subseteq I^{'}$, the following formulation presents a sub-problem of (LP-MM) consisting of all pairs in I and some pairs in $I^{'}$.

$$\max \sum_{(i,j)\in I} c_{ij}(1 - x_{ij}) + \sum_{(i,j)\in \mathcal{I}} c_{ij}(1 - x_{ij}) \tag{LPs-MM($\mathcal{I}$)}$$

$$x_{ij} + x_{jk} - x_{ik} \geq 0 \quad \forall (i, j), (j, k), (i, k) \in I \cup \mathcal{I}, c_{ij} \geq 0 \vee c_{jk} \geq 0 \tag{12}$$

$$x_{ij} - x_{jk} + x_{ik} \geq 0 \quad \forall (i, j), (j, k), (i, k) \in I \cup \mathcal{I}, c_{ij} \geq 0 \vee c_{ik} \geq 0 \tag{13}$$

$$-x_{ij} + x_{jk} + x_{ik} \geq 0 \quad \forall (i, j), (j, k), (i, k) \in I \cup \mathcal{I}, c_{jk} \geq 0 \vee c_{ik} \geq 0 \tag{14}$$

$$x_{ij} \in [0, 1] \quad \forall\, i < j,\ (i, j), (j, k), (i, k) \in I \cup \mathcal{I} \tag{15}$$

It can be easily turned out that (LPs-MM($\emptyset$)) is the smallest formulation and (LPs-MM(I')) is equivalent to (LP-MM) itself. Please note that, since for all $(i, j) \in \mathcal{I} \subseteq I'$ we have $c_{ij} \leq 0$, (LPs-MM($\mathcal{I}$)) clearly provides an upper bound of the optimal value of (LP-MM), and moreover, adding variables can never worsen the upper bound. Furthermore, the following theorem brings forward a condition under which the upper bound is equal to the optimal value of (LP-MM).

Theorem 3.1: If an optimal solution $\bar{x}^* = (\bar{x}^*_{ij})_{(i,j)\in I\cup\mathcal{I}}$ to (LPs-MM($\mathcal{I}$)) satisfies the condition $(*)$, then $(x^*_{ij})_{(i,j)\in I_{all}}$ is an optimal solution to (LP-MM), where

$$x^*_{ij} = \begin{cases} \bar{x}^*_{ij} & ;\ (i,j) \in I \cup \mathcal{I} \\ 1 & ;\ \text{otherwise} \end{cases} \tag{16}$$

and

$$(*) \begin{cases} \bar{x}^*_{ij} + \bar{x}^*_{jk} \geq 1;\ (i,j),(j,k) \in I \cup \mathcal{I}, c_{ij} \geq 0 \vee c_{jk} \geq 0, (i,k) \in I' - \mathcal{I} \\ \bar{x}^*_{ij} + \bar{x}^*_{ik} \geq 1;\ (i,j),(i,k) \in I \cup \mathcal{I}, c_{ij} \geq 0 \vee c_{ik} \geq 0, (j,k) \in I' - \mathcal{I} \\ \bar{x}^*_{jk} + \bar{x}^*_{ik} \geq 1;\ (j,k),(i,k) \in I \cup \mathcal{I}, c_{jk} \geq 0 \vee c_{ik} \geq 0, (i,j) \in I' - \mathcal{I} \end{cases}$$

Proof. Suppose that $\bar{x}^* = (\bar{x}^*_{ij})_{(i,j)\in I\cup\mathcal{I}}$ is an optimal solution to (LPs-MM($\mathcal{I}$)) that satisfies the condition $(*)$. Let $x^* = (x^*_{ij})_{(i,j)\in I_{all}}$, such that for every $(i,j) \in I_{all}$, (x^*_{ij}) is defined by Equation (16). We indicate that x^* is an optimal solution to (LP-MM). First of all, x^* is feasible for (LP-MM). To prove that, we suffice to confirm that the first set of constraints of (LP-MM) (eq. (8)) is satisfied. The same argument can be expressed for the remaining two sets of constraints. It thereby needs to be determined that for all $i < j < k$ such that $c_{ij} \geq 0$ or $c_{jk} \geq 0$, we have $x^*_{ij} + x^*_{jk} - x^*_{ik} \geq 0$. Note that eight conditions may happen to the pairs (i,j), (j,k), and (i,k). If (i,j), (j,k), $(i,k) \in I \cup \mathcal{I}$, the constraints are satisfied because they are also in (LPs-MM($\mathcal{I}$)). If (i,j), $(j,k) \in I \cup \mathcal{I}$ and $(i,k) \in I' - \mathcal{I}$, the constraints are again satisfied due to the condition $(*)$. Furthermore, in the remaining cases, at least one of x^*_{ij} or x^*_{jk} equals 1, so the constraints are again satisfied. Hence, x^* is feasible for (LP-MM). As a result, it is enough to show that the objective value of x^* in (LP-MM) is equal to that of $\bar{x}^*$ in (LPs-MM($\mathcal{I}$)). Point out that one can rewrite the objective function of (LP-MM) as follows:

$$\underbrace{\sum_{(i,j)\in I} c_{ij}(1 - x_{ij}) + \sum_{(i,j)\in\mathcal{I}} c_{ij}(1 - x_{ij})}_{F} + \underbrace{\sum_{(i,j)\in I_{all} - I - \mathcal{I}} c_{ij}(1 - x_{ij})}_{Z}.$$

F is exactly the objective value of (LPs-MM($\mathcal{I}$)), and Z equals 0 according to Equation (16). Therefore, the objective value of x^* in (LP-MM) is equal to that of $\bar{x}^*$ in (LPs-MM($\mathcal{I}$)). □

Based on the above discussion, we state the following scheme for obtaining the optimal solution to (LP-MM).

- Start solving (LPs-MM($\mathcal{I}$)) with $\mathcal{I} = \emptyset$ and adding those $x_{ij} \in I' - \mathcal{I}$ that violate inequalities in $(*)$ in each iteration, until an optimal solution to (LPs-MM($\mathcal{I}$)) satisfies $(*)$.
- Employing a row generation for solving (LPs-MM($\mathcal{I}$)) in each repeat.

B. Rounding algorithm

Recall from what we discussed in Section II-A that a solution x^* to (LP-MM) expresses the *LP distance* such that the lower the x^*_{ij}, the more tendency the nodes i and j have to be in a same community. Our local search-based procedure rounds the distance between vertices (or, as we will see, move the vertices among communities) based on simultaneously using the LP distance and the value of (IP-MM) [1]. Assume that $x^* = (x^*_{ij})_{(i,j)\in V^2}$ is an optimal fractional solution to (LP-MM). We denote each $C \subset V$ a *community* if we have $x^*_{ij} = 0$ for every $i, j \in C$. Further, by *assigning* node i we mean to round down x^*_{ij} to 0 for every $j \in C$ and round it up to 1; otherwise, where C is the community whose *center node* (explained in the next paragraph) has the minimum distance from i (w.r.t the LP distance).

The main idea of the local search-based rounding procedure is to obtain the best communities leading to the maximum possible value of (IP-MM) by wisely assigning nodes. Intuitively, the algorithm starts from an initial solution, which is the set of communities achieved by the optimal solution x^* to (LP-MM), and then iteratively moves to the neighbor solutions. In brief, it starts with randomly associating a *center node* for each community and then assigning each node j, which is not a member of any community. Next, it computes the value of (IP-MM)[2]. Afterward, it iteratively greedily improves the center vertices based on one of the three functions *Add*, *Delete*, and *Swap* at a time. Then it updates communities by reassigning every node. To be more precise, in each repeat, *Add* and *Delete* functions respectively check whether adding or deleting a center node (and therefore, the corresponding community) can make any progress in the value of (IP-MM) and if that so, the best action leading to this improvement will be recorded. On the other hand, the function *Swap* tries to discover the best switch between a non-center and a center node that leads to the maximum improvement in (IP-MM) value. At last, the best function leading to the best gain in the value of (IP-MM) will be selected, and in this way, communities will be updated. The above procedure will be repeated until the best possible community structures regarding the obtained value of (IP-MM) are found. We note that in the case that solving (LP-MM) does not lead to obtaining any communities at the very beginning (i.e., $x^*_{ij} \neq 0$ for every $i, j \in V$ such that $i \neq j$), the algorithm randomly chooses a number $k \in \{1, 2, \ldots, n\}$ of nodes as center vertices and assigns each of the remaining vertices. Algorithm 1 elaborates the pseudo-code of this technique.

IV. Computational results

In this section, we present a performance evaluation for our proposed method by using 12 commonly-used and well-known real-world networks that are listed in Table I. Ground truth (i.e., the optimal community structures) is available and known for each of these networks, and therefore, one can facilely measure the quality of a community detection algorithm by estimating the similarities between the communities obtained by the algorithm and the ground truth. For doing this, we use the well-known performance metric NMI.

A. Normalized Mutual Information (NMI)

NMI [50] is indeed a well-known clustering comparison metric. Nevertheless, it can perfectly evaluate the similarity

[1] By a *value of (IP-MM)*, we mean the value of the objective function of (IP-MM) with respect to an integer solution.

[2] Note that, after assigning all vertices, we have integer solution.

Algorithm 1: Local search-based rounding procedure.

```
Input: $x^* = (x^*_{ij})_{(i,j)\in V^2}$ // an optimal solution to (LP-MM).
Output: set of communities $\mathcal{C}$ of the network $G$.
let $T = \{T_1, T_2, \ldots, T_k\}$ be the set of $k$ initial communities obtained by
  $x^*$;
if $|T| \neq \emptyset$ then
    let $S = \{\mu_1, \mu_2, \ldots, \mu_k\}$ such that $\mu_i$ is a randomly selected
      member of $T_i$, for all $i \in \{1, 2, \ldots, k\}$;
else
    let $S = \{\mu_1, \mu_2, \ldots, \mu_k\}$ be a set of $k$ randomly chosen vertices, for
      a random $k \in \{2, \ldots, n\}$;
$(\mathcal{C}, Q) \leftarrow$CalculateGain($S$);
$Q_{temp} \leftarrow 0$;
while $Q > (1+\epsilon)Q_{temp}$ do
    // small constant $\epsilon$ guarantees that running time remains polynomial. See
      [37], [38].
    $Q_{temp} \leftarrow Q$;
    $(\mathcal{C}, Q, S) \leftarrow BestMove(S)$;
Return $(\mathcal{C})$;
− − − − − − − − − − − − − − − − − − − − − − − − − − − − − −
// Functions declaration:
CalculateGain $(S)$
    let $C_i = \{\mu_i\}$, for every $\mu_i \in S$ and $1 \leq i \leq |S|$;
    for every $i \in V - S$ do
        assign $i$; (i.e., $C_j \leftarrow C_j \cup \{i\}$ where
          $j = argmin\{x^*_{i\mu_j} : 1 \leq j \leq |S|\}$)
    $\mathcal{C} \leftarrow \{C_1, C_2, \ldots, C_k\}$;
    $Q \leftarrow$ the value of (IP-MM) w.r.t the set of communities $\mathcal{C}$;
    Return $(\mathcal{C}, Q)$;
BestMove $(S)$
    $(S^{add}, \mathcal{C}^{add}, Q^{add}) \leftarrow$ Add(S);
    $(S^{delete}, \mathcal{C}^{delete}, Q^{delete}) \leftarrow$ Delete(S);
    $(S^{swap}, \mathcal{C}^{swap}, Q^{swap}) \leftarrow$ Swap(S);
    retrieve the highest (IP-MM) value $Q$, the best set of communities $\mathcal{C}$,
      and the best set of center nodes $S$;
    Return $(\mathcal{C}, Q, S)$;
Add $(S)$
    for every $i \in V - S$ do
        $S^{add} \leftarrow S \cup \{i\}$;
        $(\mathcal{C}^{add}, Q^{add}) \leftarrow$ CalculateGain($S^{add}$);
        remember current $S^{add}$, $\mathcal{C}^{add}$, and $Q^{add}$;
    Return $(S^{add}$, $\mathcal{C}^{add}$, $Q^{add})$ corresponding to the highest obtained
      $Q^{add}$;
Delete $(S)$
    for every $i \in S$ do
        $S^{delete} \leftarrow S - \{i\}$;
        $(\mathcal{C}^{delete}, Q^{delete}) \leftarrow$ CalculateGain($S^{delete}$);
        remember current $S^{delete}$, $\mathcal{C}^{delete}$, and $Q^{delete}$;
    Return $(S^{delete}$, $\mathcal{C}^{delete}$, $Q^{delete})$ corresponding to the highest
      obtained $Q^{delete}$;
Swap $(S)$
    for every $i \in S$ do
        for every $j \in V - S$ do
            $S^{swap} \leftarrow (S - \{i\}) \cup \{j\}$;
            $(\mathcal{C}^{swap}, Q^{swap}) \leftarrow$ CalculateGain($S^{swap}$);
            remember current $S^{swap}$, $\mathcal{C}^{swap}$, and $Q^{swap}$;
    Return $(S^{swap}$, $\mathcal{C}^{swap}$, $Q^{swap})$ corresponding to the highest
      $Q^{swap}$;
```

between the optimal communities and those discovered by an algorithm. Suppose that for a given network G, $\mathcal{C}(\mathcal{A}) = \{C_1, \ldots, C_k\}$ and $\mathcal{C}' = \{C'_1, \ldots, C'_{k'}\}$ be respectively a set of communities obtained by an algorithm $\mathcal{A}$ and the ground truth. The NMI value corresponding to the algorithm $\mathcal{A}$ can be written as

$$NMI = \frac{-2\sum_{x=1}^{|\mathcal{C}|}\sum_{y=1}^{|\mathcal{C}'|} \frac{|C_x \cap C'_y|}{n} log(\frac{n|C_x \cap C'_y|)}{|C_x||C'_y|}}{\sum_{x=1}^{|\mathcal{C}|} \frac{C_x}{n} log(\frac{C_x}{n}) + \sum_{y=1}^{|\mathcal{C}'|} \frac{C'_y}{n} log(\frac{C'_y}{n})} \tag{17}$$

TABLE I: Networks under-study

ID	Network	n	m
1	Zachary's karate club [39]	34	78
2	Mexican Politicians [40]	35	117
3	Dolphin network [41]	62	159
4	Les Miserables [41]	77	254
5	p53 protein [42]	104	226
6	Books about U.S. politics [43]	105	441
7	American college football [44]	115	613
8	Citation graph drawing [45]	311	640
9	USAir97 [46]	332	2126
10	C. Elegans [47]	453	2025
11	Erdos collaboration [48]	472	1314
12	Electronic circuit [49]	512	819

In the case where the detected communities are identical to the ground truth, the NMI takes its maximum value one, while in the case where the two sets totally disagree, the NMI score is zero. Generally, the more the NMI, the better community structures have been found.

B. Experiments

In what follows, we provide a complete evaluation that shows how our relation matrix or/and rounding technique can individually resp. together improve the old-fashion relation matrix or/and other rounding procedures and the conventional Max-Min Modularity algorithm.

All tests are conducted on a computer system with a processor Intel(R) Core(TM) i5-7300U CPU @ 2.60GHz, 2712 Mhz, 2 Core(s), 4 Logical Processor(s), 8 GB of Rams, and Win10 OS. Algorithms are implemented with C++, and CPLEX optimizer 12.9 is used for solving linear programming.

Fig. 2 provides a comprehensive comparison by evaluating communities that are discovered based on the following cases:

- Our proposed method (the blue diagram): Using the relation matrix, proposed in Section II-A, to model the Max-Min Modularity Maximization problem, solving (LP-MM) via the row/column generation method introduced in Section III-A, and detecting communities (obtaining integer solutions) by the devised rounding technique (Section III-B).
- Replacing the relation matrix suggested in [21] with our relation matrix but using the hierarchical algorithm proposed in [21] (the gray diagram).
- Applying the relation matrix introduced in [21] to model the Max-Min Modularity Maximization problem and using our proposed rounding procedure to obtain communities (the yellow diagram).
- Using our relation matrix and row/column generation technique to solve (LP-MM), but employing the rounding algorithm proposed by Agarwal and Kempe [32] [3] instead of our rounding procedure; (The black diagram).

[3] The authors of [32] introduced a rounding procedure to obtain the integer solution to the Modularity Maximization problem. Their method is actually derived from a rounding procedure that is originally proposed for the correlation clustering problem. However, it led to raising unwarranted singleton and a number of low-quality communities that made them apply a series of Kernighan-Lin shifts [51] to improve community structures. Here we used their technique to rounds (LP-MM) solution.

- Applying our rounding method to the optimal solution to the linear programming relaxation of the Modularity Maximization problem obtained in [34] (green diagram).
- Max-Min Modularity, proposed in [21] (red diagram).

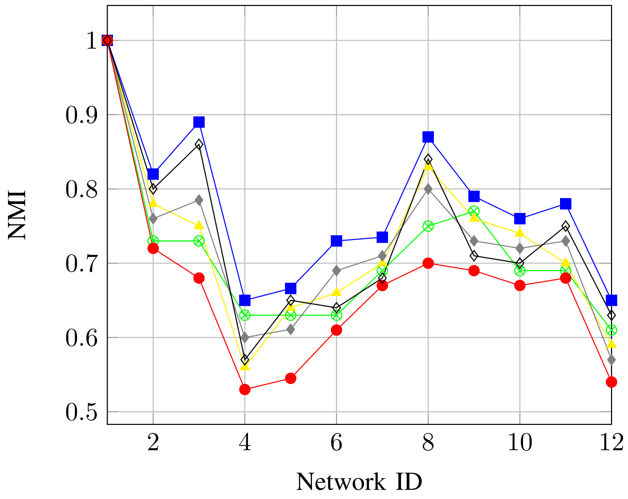


Fig. 2: Comparison between NMI values achieved by (i) blue curve: our method, (ii) gray curve: using our proposed relation matrix but the old-fashion algorithm, (iii) yellow curve: using the old-fashion relation matrix, but applying our proposed rounding procedure, (iv) green curve: applying our rounding method to the optimal solution to the linear programming relaxation of the Modularity Maximization problem, (v) black curve: applying our relation matrix but using another rounding algorithm to find communities. and (vi) red curve: conventional Max-Min modularity.

One can obviously conclude that the best results are achieved when the proposed method, including the new relations matrix and also the devised rounding algorithm, is used. In particular, comparing the blue and green diagrams shows the advantage of solving (linear relaxation) of the Max-Min Modularity Maximization problem rather than solving the (linear relaxation) of the Modularity Maximization problem. On the other hand, the worst result occurs when we just use the Max-Min Modularity proposed in [21]. So, an immediate consequence might be that while the idea behind the Max-Min Modularity is so clever and interesting, the relation matrix and also the hierarchical algorithm introduced in [21] do not lead to a very high-quality result.

As we already mentioned in Section II-A, comparing the gray and red diagrams can show us the superiority of using our proposed relation matrix instead of the one introduce in [21]. On the other hand, by considering the blue and black diagrams, one can recognize the preponderance of the proposed rounding algorithm over the famous rounding procedure suggested in [32]. A final remark might be that although the proposed relation matrix and the developed rounding technique alone improves the results, the high efficiency of the method considerably relies on their simultaneous application. It means that, for example, applying our proposed rounding algorithm but using the traditional relation matrix cannot always lead the promising results; See the yellow diagram.

V. Conclusion

In this work, we first introduced a systematic way to generate a more accurate relation matrix for the Max-Min Modularity Maximization problem based on the optimal solution to the linear relaxation programming of the Modularity Maximization problem. After that, according to this new relation matrix, we modeled the standard integer formulation for the Max-Min Modularity Maximization problem and employed a row/column generation technique to solve its linear relaxation version. We also devised a local search-based rounding method that facilitates us to round fractional solutions to integer ones and detect communities of a network in a very accurate way. The proposed computational experiments showed that our results highly resemble the optimal solutions and that our algorithm outperforms the previous well-known algorithms.

References

[1] L. Jiang, L. Shi, L. Liu, J. Yao, and M. A. Yousuf, "User interest community detection on social media using collaborative filtering," *Wireless Networks*, pp. 1–7, 2019.

[2] M. Wang, C. Wang, J. X. Yu, and J. Zhang, "Community detection in social networks: an in-depth benchmarking study with a procedure-oriented framework," *Proceedings of the VLDB Endowment*, vol. 8, no. 10, pp. 998–1009, 2015.

[3] Y. Atay, I. Koc, I. Babaoglu, and H. Kodaz, "Community detection from biological and social networks: A comparative analysis of metaheuristic algorithms," *Applied Soft Computing*, vol. 50, pp. 194–211, 2017.

[4] D. Krioukov, M. Kitsak, R. S. Sinkovits, D. Rideout, D. Meyer, and M. Boguñá, "Network cosmology," *Scientific reports*, vol. 2, p. 793, 2012.

[5] S. Aparicio, J. Villazón-Terrazas, and G. Álvarez, "A model for scale-free networks: application to twitter," *Entropy*, vol. 17, no. 8, pp. 5848–5867, 2015.

[6] N. Tremblay and P. Borgnat, "Graph wavelets for multiscale community mining," *IEEE Transactions on Signal Processing*, vol. 62, no. 20, pp. 5227–5239, 2014.

[7] O. A. Linares, G. M. Botelho, F. A. Rodrigues, and J. B. Neto, "Segmentation of large images based on super-pixels and community detection in graphs," *IET Image Processing*, vol. 11, no. 12, pp. 1219–1228, 2017.

[8] L. M. Freitas and M. G. Carneiro, "Community detection to invariant pattern clustering in images," in *2019 8th Brazilian Conference on Intelligent Systems (BRACIS)*. IEEE, 2019, pp. 610–615.

[9] F. D. Malliaros and M. Vazirgiannis, "Clustering and community detection in directed networks: A survey," *Physics Reports*, vol. 533, no. 4, pp. 95–142, 2013.

[10] M. E. Newman, "Modularity and community structure in networks," *Proceedings of the national academy of sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.

[11] U. Brandes, D. Delling, M. Gaertler, R. Gorke, M. Hoefer, Z. Nikoloski, and D. Wagner, "On modularity clustering," *IEEE transactions on knowledge and data engineering*, vol. 20, no. 2, pp. 172–188, 2007.

[12] P. Schuetz and A. Caflisch, "Efficient modularity optimization by multistep greedy algorithm and vertex mover refinement," *Physical Review E*, vol. 77, no. 4, p. 046112, 2008.

[13] S. Cafieri, A. Costa, and P. Hansen, "Reformulation of a model for hierarchical divisive graph modularity maximization," *Annals of Operations Research*, vol. 222, no. 1, pp. 213–226, 2014.

[14] B. Rajita and S. Panda, "Community detection techniques for evolving social networks," in *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE, 2019, pp. 681–686.

[15] A. Ferdowsi and A. Abhari, "Generating high-quality synthetic graphs for community detection in social networks," in *2020 Spring Simulation Conference (SpringSim)*. IEEE, 2020, pp. 1–10.

[16] R. Guimera and L. A. N. Amaral, "Functional cartography of complex metabolic networks," *nature*, vol. 433, no. 7028, pp. 895–900, 2005.

[17] D. Aloise, S. Cafieri, G. Caporossi, P. Hansen, S. Perron, and L. Liberti, "Column generation algorithms for exact modularity maximization in networks," *Physical Review E*, vol. 82, no. 4, p. 046112, 2010.

[18] G. Xu, S. Tsoka, and L. G. Papageorgiou, "Finding community structures in complex networks using mixed integer optimisation," *The European Physical Journal B*, vol. 60, no. 2, pp. 231–239, 2007.

[19] S. Fortunato and M. Barthelemy, "Resolution limit in community detection," *Proceedings of the national academy of sciences*, vol. 104, no. 1, pp. 36–41, 2007.

[20] T. Chakraborty, A. Dalmia, A. Mukherjee, and N. Ganguly, "Metrics for community analysis: A survey," *ACM Computing Surveys (CSUR)*, vol. 50, no. 4, pp. 1–37, 2017.

[21] J. Chen, O. R. Zaïane, and R. Goebel, "Detecting communities in social networks using max-min modularity," in *Proceedings of the 2009 SIAM international conference on data mining*. SIAM, 2009, pp. 978–989.

[22] J. Scripps, P.-N. Tan, and A.-H. Esfahanian, "Exploration of link structure and community-based node roles in network analysis," in *Seventh IEEE international conference on data mining (ICDM 2007)*. IEEE, 2007, pp. 649–654.

[23] J. Duch and A. Arenas, "Community detection in complex networks using extremal optimization," *Physical review E*, vol. 72, no. 2, p. 027104, 2005.

[24] T. Richardson, P. J. Mucha, and M. A. Porter, "Spectral tripartitioning of networks," *Physical Review E*, vol. 80, no. 3, p. 036111, 2009.

[25] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, vol. 69, no. 2, p. 026113, 2004.

[26] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, p. P10008, 2008.

[27] R. Guimera and L. A. N. Amaral, "Cartography of complex networks: modules and universal roles," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2005, no. 02, p. P02001, 2005.

[28] S. Boccaletti, M. Ivanchenko, V. Latora, A. Pluchino, and A. Rapisarda, "Detecting complex network modularity by dynamical clustering," *Physical Review E*, vol. 75, no. 4, p. 045102, 2007.

[29] A. Tsitsulin, J. Palowitch, B. Perozzi, and E. Müller, "Graph clustering with graph neural networks," *arXiv preprint arXiv:2006.16904*, 2020.

[30] C. Shi, Y. Liu, and P. Zhang, "Weighted community detection and data clustering using message passing," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2018, no. 3, p. 033405, 2018.

[31] Y. Q. Niu, B. Q. Hu, W. Zhang, and M. Wang, "Detecting the community structure in complex networks based on quantum mechanics," *Physica A: Statistical Mechanics and Its Applications*, vol. 387, no. 24, pp. 6215–6224, 2008.

[32] G. Agarwal and D. Kempe, "Modularity-maximizing graph communities via mathematical programming," *The European Physical Journal B*, vol. 66, no. 3, pp. 409–418, 2008.

[33] T. N. Dinh and M. T. Thai, "Finding community structure with performance guarantees in complex networks," *arXiv preprint arXiv:1108.4034*, 2011.

[34] A. Miyauchi and Y. Miyamoto, "Computing an upper bound of modularity," *The European Physical Journal B*, vol. 86, no. 7, p. 302, 2013.

[35] M. E. Newman, "Analysis of weighted networks," *Physical review E*, vol. 70, no. 5, p. 056131, 2004.

[36] A. Miyauchi and N. Sukegawa, "Redundant constraints in the standard formulation for the clique partitioning problem," *Optimization Letters*, vol. 9, no. 1, pp. 199–207, 2015.

[37] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit, "Local search heuristics for k-median and facility location problems," *SIAM Journal on computing*, vol. 33, no. 3, pp. 544–562, 2004.

[38] A. Gupta and K. Tangwongsan, "Simpler analyses of local search algorithms for facility location," *arXiv preprint arXiv:0809.2554*, 2008.

[39] W. W. Zachary, "An information flow model for conflict and fission in small groups," *Journal of anthropological research*, vol. 33, no. 4, pp. 452–473, 1977.

[40] J. Gil-Mendieta and S. Schmidt, "The political network in mexico," *Social Networks*, vol. 18, no. 4, pp. 355–381, 1996.

[41] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations," *Behavioral Ecology and Sociobiology*, vol. 54, no. 4, pp. 396–405, 2003.

[42] R. Guimera, L. Danon, A. Diaz-Guilera, F. Giralt, and A. Arenas, "Self-similar community structure in a network of human interactions," *Physical review E*, vol. 68, no. 6, p. 065103, 2003.

[43] A. Mahajan and M. Kaur, "Various approaches of community detection in complex networks: a glance," *International Journal of Information Technology and Computer Science (IJITCS)*, vol. 8, no. 35, 2016.

[44] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the national academy of sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[45] N. Meghanathan, "A greedy algorithm for neighborhood overlap-based community detection," *Algorithms*, vol. 9, no. 1, p. 8, 2016.

[46] V. Batagelj and A. Mrvar, "Pajek datasets (2006)," 2009.

[47] A. Cangelosi and D. Parisi, "A neural network model of caenorhabditis elegans: the circuit of touch sensitivity," *Neural processing letters*, vol. 6, no. 3, pp. 91–98, 1997.

[48] V. Batagelj and A. Mrvar, "Pajek." 2014.

[49] S. Chand and S. Mehta, "Community detection using nature inspired algorithm," in *Hybrid Intelligence for Social Networks*. Springer, 2017, pp. 47–76.

[50] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2005, no. 09, p. P09008, 2005.

[51] B. W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *The Bell system technical journal*, vol. 49, no. 2, pp. 291–307, 1970.

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 337–340

DOI: 10.15439/2021F62
ISSN 2300-5963 ACSIS, Vol. 25

Short Performance Analysis of the LTE and 5G Access Technologies in NS-3

Maroš Baumgartner, Jozef Juhár, Ján Papaj
Technical University of Košice
Faculty of Electrical Engineering and Informatics
Slovak Republic
Email: {maros.baumgartner jozef.juhar, jan.papaj}@tuke.sk

***Abstract*—Nowadays, the requirements for data transmission and efficiency of IoT networks are increasing. Network efficiency at all levels can be increased by using 5G networks. In this paper, we simulate and analyse the LTE, enhanced Mobile BroadBand, and enhanced Mobile BroadBand with Millimeter-wave services in scenarios with different numbers of IoT nodes and analyze the energy consumption and energy efficiency achieved results. Simulated scenarios and results were obtained using NS-3. Energy and mmWave frameworks were analyzed in this work because they were *the* main part of the research.**

***Index Terms*—Simulation, 5G, LTE, NS-3, eMBB, mmWave.**

I. Introduction

NEXT-GENERATION networks, known as 5G networks, are becoming an increasingly important part of our lives. These networks should bring improvements in various areas of data transmission, such as capacity, end-to-end latency, robustness, scalability, data transfer speed and energy efficiency.

5G enables faster, more stable, and more secure connectivity that's advancing everything from self-driving vehicles, to smart grids for renewable energy, to AI-enabled robots on factory floors. All these static and dynamic sensors acquire a large amount of IoT data, which is used to communicate and improve network efficiency.

One of the essential parameters in 5G networks and IoT (Internet of Things) is the energy efficiency and energy consumption of individual IoT devices. In addition to reducing network operating costs, the main goal of network energy efficiency is to reduce energy consumption and extend battery life in IoT end devices, such as mobile phones, laptops, drones, and the like [1].

Because LTE (Long Term Evolution) is currently the most widespread data transmission service. Therefore, in this article, we focused on comparing the achieved results of energy efficiency and energy consumption of older LTE networks with 5G network technologies in the form of enhanced Mobile BroadBand (eMBB) and millimeter waves (mmWave) [10].

The article is divided into several parts. In the first part, we describe the implemented mmWave model in NS-3 (Network Simulator 3) used in the simulations. This part also describes the energy model with which we obtained and analyzed the results [2].

The following section describes the simulation scenarios, the used parameters, and the achieved results. The average values of energy efficiency and energy consumption of IoT nodes of LTE service were compared with the results obtained by simulation of 5G networks.

II. NS-3 frameworks

The following section describes the mmWave and energy model for the NS-3 network simulator.

A. mmWave model

The basis of the mmWave module for NS-3 is the LENA model. He is considered the most robust. It was designed to simulate complex mobile networks in the style of 3GPP (3rd Generation Partnership Project) [1]. In addition to LTE / EPC (Evolved Packet Core) protocols, it also implements its own MAC and PHY layers [2][3]. These layers have been described in detail by Rebato and Mezzavilla in their work [4][5].

The basic UML diagram of the mmWave module, which describes the relationships between classes and layers, is shown in Fig. 1. The whole diagram represents the end-to-end structure of the simulator.

The MmWaveEndNetDevice and MmWaveUeNetDevice classes have the function of radio bins for mmWave eNodeB and mmWave UE. In addition, the McUeNetDevice class ensures that a NetDevice device can connect to mmWave and LTE technologies using a dual stack.

The MAC, MmWaveEnbMac and MmWaveUeMac layer classes implement SAP (Service Access Point), user interface and LTE module for cooperation with the LTE RLC layer [6]. MAC, MmWaveMacScheduler, and derived classes implement support for classes from the RLC group, i.e. TM (Transparent Mode), UM (Unacknowledged Mode), SM (Saturation Mode) and AM (Acknowledge Mode). SAP

The research presented in this paper was supported by the Ministry of Education, Science, Research and Sport of the Slovak Republic under the research project VEGA 1/0753/20, KEGA 009TUKE-4/2019, and by Slovak Research and Development Agency project APVV-17- 0208.

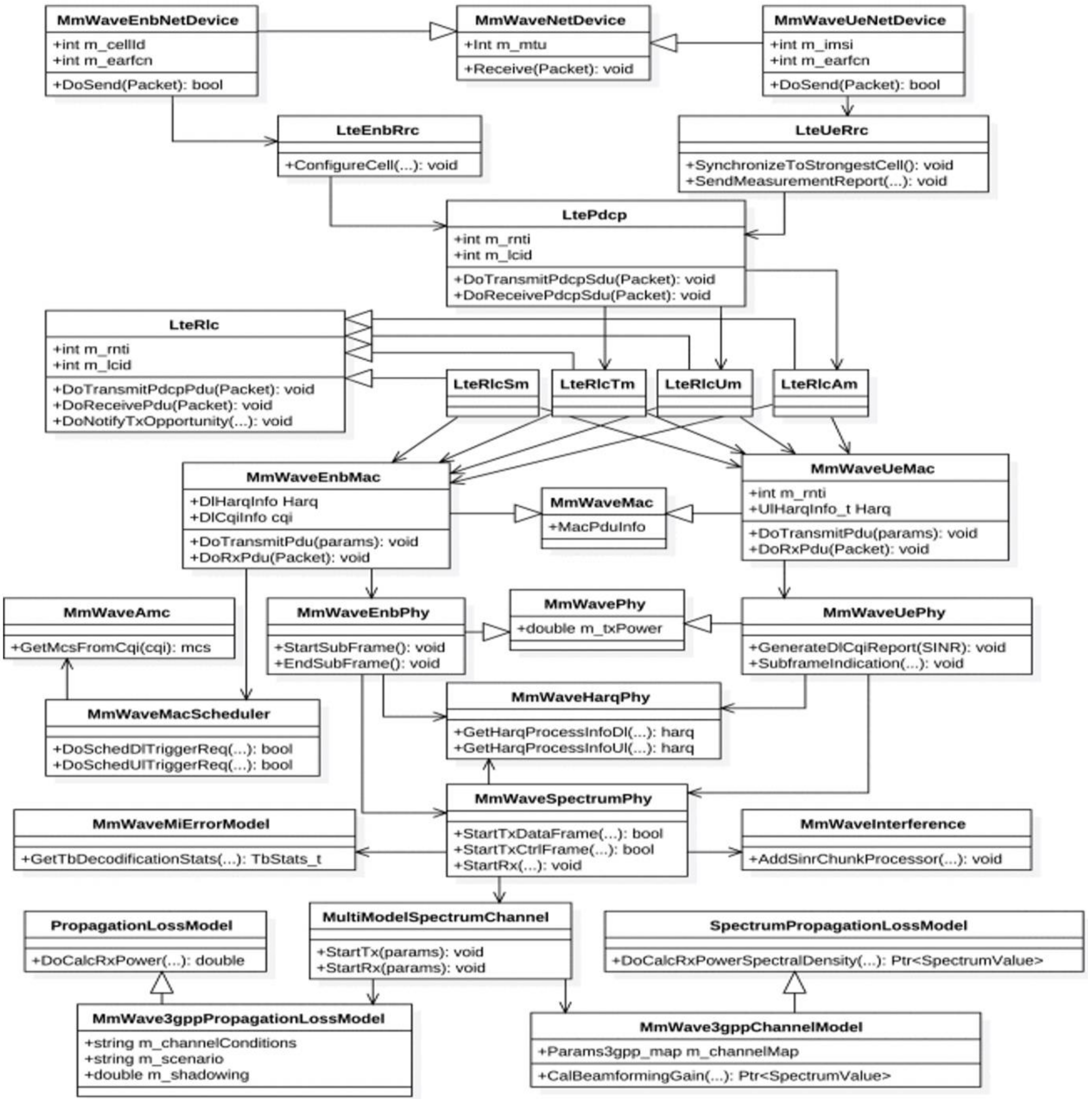

MmWaveEnbNetDevice
+int m_cellId
+int m_earfcn
+DoSend(Packet): bool
MmWaveNetDevice
+Int m_mtu
+Receive(Packet): void
MmWaveUeNetDevice
+int m_imsi
+int m_earfcn
+DoSend(Packet): bool
LteEnbRrc
+ConfigureCell(...): void
LteUeRrc
+SynchronizeToStrongestCell(): void
+SendMeasurementReport(...): void
LtePdcp
+int m_rnti
+int m_lcid
+DoTransmitPdcpSdu(Packet): void
+DoReceivePdcpSdu(Packet): void
LteRlc
+int m_rnti
+int m_lcid
+DoTransmitPdcpPdu(Packet): void
+DoReceivePdu(Packet): void
+DoNotifyTxOpportunity(...): void
LteRlcSm
LteRlcTm
LteRlcUm
LteRlcAm
MmWaveEnbMac
+DlHarqInfo Harq
+DlCqiInfo cqi
+DoTransmitPdu(params): void
+DoRxPdu(Packet): void
MmWaveMac
+MacPduInfo
MmWaveUeMac
+int m_rnti
+UlHarqInfo_t Harq
+DoTransmitPdu(params): void
+DoRxPdu(Packet): void
MmWaveAmc
+GetMcsFromCqi(cqi): mcs
MmWaveEnbPhy
+StartSubFrame(): void
+EndSubFrame(): void
MmWavePhy
+double m_txPower
MmWaveUePhy
+GenerateDlCqiReport(SINR): void
+SubframeIndication(...): void
MmWaveMacScheduler
+DoSchedDlTriggerReq(...): bool
+DoSchedUlTriggerReq(...): bool
MmWaveHarqPhy
+GetHarqProcessInfoDl(...): harq
+GetHarqProcessInfoUl(...): harq
MmWaveSpectrumPhy
+StartTxDataFrame(...): bool
+StartTxCtrlFrame(...): bool
+StartRx(...): void
MmWaveMiErrorModel
+GetTbDecodificationStats(...): TbStats_t
MmWaveInterference
+AddSinrChunkProcessor(...): void
PropagationLossModel
+DoCalcRxPower(...): double
MultiModelSpectrumChannel
+StartTx(params): void
+StartRx(params): void
SpectrumPropagationLossModel
+DoCalcRxPowerSpectralDensity(...): Ptr<SpectrumValue>
MmWave3gppPropagationLossModel
+string m_channelConditions
+string m_scenario
+double m_shadowing
MmWave3gppChannelModel
+Params3gpp_map m_channelMap
+CalBeamformingGain(...): Ptr<SpectrumValue>

Fig. 1 UML diagram of the mmWave module [6]

is implemented in the LteEnbRrc class using a MAC scheduler due to configurations at the LTE layer of Radio Resource *Control (RRC)* [6][7].

Classes from the MmWavePhy group provide directional transmission, data reception via downlink and uplink. They also take care of channels that work on the principle of control MAC messages [1]-[3]. MmWaveSpectrumPhy class instances are used to communicate Phy class instances through SpectrumChannel [6]. The MmWaveSpectrumPhy class and its instances are shared for uplink and downlink due to the physical layer of the mmWave module, which is based on TDD (Time Division Duplexing) [8][9].

B. Energy framework

Power consumption is a crucial feature for wireless IoT nodes in mobile networks. Therefore, for research on energy consumption, and energy framework was implemented in the NS-3 simulator, using which data were obtained for further processing [11]. Energy model NS-3 consists of 3 parts:

- Energy source
- Energy model
- Energy harvester

The energy source and energy model for simulation use are described below.

The energy source in the network represents the energy source in each IoT node. Each node can be connected to multiple device energy models. If an energy source is connected to such models, the specific device will obtain energy from this source [11][12][13].

The essential function of the power source is to provide power to the node. If the energy in the nodes is depleted, the node informs neighbouring nodes that can respond to this event, obtaining information about the remaining power or the battery charge level.

To simulate power supplies, the energy source class must handle two essential effects of practical batteries:

- Rate capacity effect - Decrease of battery life when the current draw is higher than the rated value of the battery.
- Recovery effect - Increase battery life when the battery is alternating between discharge and idle states [11].

The energy source class divides a node into multiple smaller devices. Each of these devices consumes energy separately. Therefore, the energy source periodically asks for the energy consumed from all devices in the same node, from which it calculates the total energy consumption.

The model for device power consumption in IoT nodes is the device energy class. Each device has several states defined, and each state is associated with a certain value of energy consumption. If the state of the device changes, the energy model notifies the energy source class of the difference in the device's current energy consumption. Based on this, the energy source calculates the current energy consumption and updates the remaining energy value. Fig. 2 shows a block diagram of the energy model in NS-3 [10] - [13].

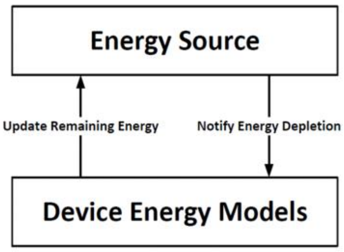


Fig. 2 Block diagram of the energy model in NS-3

III. Simulation Parameters and Scenarios

This section contains a description of the simulation parameters used, simulation scenarios and methods implemented in the NS-3 simulator.

This work focuses on simulating and comparing individual data transmission methods for research in robust data transmission using 5G networks. The work compares and analyzes the results of simulations of LTE, eMBB and eMBB with mmWave transmission methods in terms of energy efficiency and energy consumption of the network in different types of scenarios, which were created using NS-3. This type of new 5G service was chosen mainly because it has a rigorous and well-defined standard.

A. Simulation parameters

In this research, cases of different numbers of IoT end devices received by data packets were simulated and analysed. The proposed scenarios were simulated in a simulation area without obstacles of 600 m x 600 m with a User Datagram Protocol (UDP) transport protocol and with a packet size of 512 bytes. Each simulation scenario was run 200 times. The IoT nodes in the network moved with a random model of mobility in the simulated area.

Table I below lists the parameters and values that were used in the simulations.

Table I. Simulation parameters

Parameter	Value
Frequency	2,160 GHz (LTE), 6 GHz (eMBB), 30 GHz (mmWave)
Bandwidth	20 MHz
Number of IoT nodes	20, 40, 60, 80, 100
End-user's mobility	Random
End-user's speed	1-5 km/h
Size of simulating area	600 x 600 m
Simulation time	100 s
Number of the simulation run	200
Transport protocol	UDP
Packet size	512 bytes

B. Achieved results

This section describes the achieved results of simulations based on energy efficiency and energy consumption in the NS-3 simulator environment. The results were compared for LTE, eMBB and eMBB with mmWave.

Fig. 3 below shows the average values of energy consumption for all types of simulated scenarios. The graph of energy consumption as a function of the number of end-users shows that as the number of end-users increases, the energy consumption per bit increases. Thus, from the point of view of energy consumption, the services of 5G eMBB and mmWave networks are more economical than the older LTE service in all types of simulated scenarios.

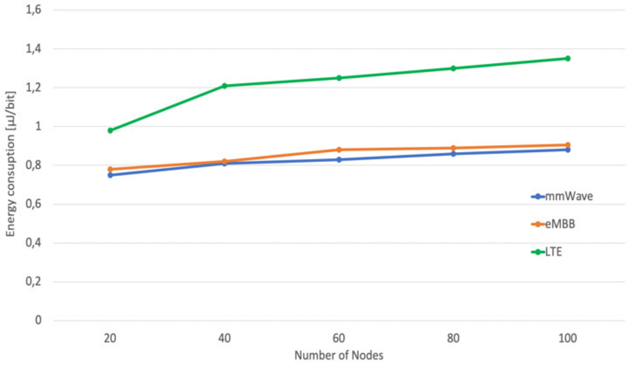


Fig. 3 Average values of energy consumption

We can see that the average values of consumed energy of LTE service range from 0.98 µJ / bit to 1.38 µJ / bit, depending on the number of end-users. Compared to eMBB and eMBB services with mmWave, this is significantly more. The eMBB service ranged from 0.76 µJ / bit to 0.905 µJ / bit. The mmWave service reached similar values, from 0.72 µJ / bit to 0.88 µJ / bit. All average values of energy consumed are shown in Table II below.

Table II. Average values of energy consumption in mJ / bit

	20 End users	40 End users	60 End users	80 End users	100 End users
LTE	0,98	1,21	1,25	1,3	1,35
eMBB	0,78	0,82	0,88	0,89	0,905
mmWave	0,75	0,81	0,829	0,86	0,88

If we compare the energy consumption of eMBB and mmWave to LTE, we find that eMBB consumed 29.81% and mmWave 32.22% with 0,05 std value (Standard Deviation) less energy than LTE.

In terms of the energy efficiency of the network, the results achieved were very similar. In Fig. 4, the older LTE service has lower energy efficiency than the eMBB and mmWave services. The average values ranged from 17.75 Mbit / J to 11.42 Mbit / J. The eMBB service ranged from 33.45 Mbit / J to 13.53 Mbit / J. The eMBB service with mmWave achieved on average 1 Mbit / J higher than the eMBB service in all simulated scenarios.

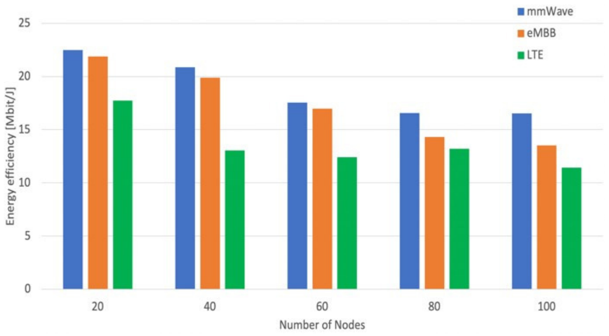


Fig. 4 Average values of energy efficiency

All achieved energy efficiency results are shown in Table III below.

Table III. Average values of energy efficiency in Mbit / J

	20 End users	40 End users	60 End users	80 End users	100 End users
LTE	17,75	13,06	12,43	13,22	11,42
eMBB	21,89	19,88	16,98	14,33	13,53
mmWave	22,49	20,87	17,56	16,58	16,54

IV. Conclusions and Future Work

This work deals with communication using millimeter waves in 5G networks and a comparison of this service with eMBB and LTE services in energy efficiency and energy consumption of IoT nodes.

In this study, we focused on mmWave because, in the future, mmWave will have the most significant impact on transmission networks in terms of bandwidth, transmission speed and energy efficiency compared to eMBB or the older LTE service.

Based on simulations of 4G and 5G networks for the UDP service, the achieved results of energy efficiency and energy consumption of IoT nodes in the network were analyzed. Individual scenarios were simulated using an NS-3 simulator into which mmWave and the energy model were implemented. Based on the achieved results, it can be said that the average values of energy efficiency and energy consumed for 5G networks are many times better than 4G networks. This means that 5G networks are proving to be more efficient in terms of transmission speed, channel capacity, or data transmission efficiency and energy efficiency for IoT.

The achieved results will form a theoretical basis in future work. We will focus on using machine learning algorithms to increase the robustness of the 5G network, which will be experimentally simulated in natural conditions using intelligent devices.

References

[1] N. Baldo, M. Miozzo, M. Requena-Esteso and J. Nin-Guerrero, "An open-source product-oriented LTE network simulator based on ns-3", Proc. 14th ACM Int. Conf. Model. Anal. Simulat. Wireless Mobile Syst., pp. 293-298, 2011, [online] Available: http://doi.acm.org/10.1145/2068897.2068948, in press.

[2] Mezzavilla M., Zhang M., Poese M., Ford R., Dutta S., Rangan S., Zorzi M., "End-to-End Simulation of 5G mmWave Networks", 2018 IEEE Communications Surveys & Tutorials.

[3] H. Tazaki et al., "Direct code execution: Revisiting library OS architecture for reproducible network experiments", Proc. 9th ACM Conf. Emerg. Netw. Exp. Technol. (CoNEXT), pp. 217-228, 2013, [online] Available: http://doi.acm.org/10.1145/2535372.2535374, in press.

[4] M. Rebato, M. Zorzi, "Simulation analysis of algorithms for interference management in 5G cellular networks using spatial spectrum sharing", Universita Degli Studi di Padova, pp. 17-33, 2015.

[5] M. Mezzavilla, S. Dutta, M. Zhang, M. R. Akdeniz, S. Rangan, "5G mmWave Module for ns-3 Network Simulator", NYU Polytechnic School of Engineering, New York, June 2015, in press.

[6] Edoardo B., "Design and performance evaluation of mm-wave vehicular networks", Politechnico di Torino, pp. 38-44, March 2019.

[7] Argha S., Abhijit M., Basabdatta P., Jay J., Krishna P., Sandip Ch., "An ns3-based Energy Module of 5G NR User Equipments for Millimeter Wave Networks", 2021 IEEE Conference on Computer Communications Workshops.

[8] Zeman K., Masek P., Stusek M., Hosek J., Silhavy P., "Accuracy Comparison of Propagation Models for mmWave Communication in NS-3", 2017 9th International Congress on Ultra Modern Telecomunications and Control Systems and Workshops (ICUMT).

[9] Zhang M., Mezzavilla M., Ford R., Rangan S., Panwar S., Mellios E., Kong D., Nix A., Zorzi M., "Transport Layer Performance in 5G mmWave Cellular", 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), San Francisco, USA.

[10] Yi S., Chun S., Lee Y., Park S.,, Jung S., "Overview of LTE and LTE-Advanced New Features", 2013 Wiley Telecom, p. 151-158.

[11] Finnegan J., Brown S., Farrell R., "Modeling the Energy Consumption of LoRaWAN in ns-3 Based on Real World Measurements", 2018 Global Information Infrastructure and Networking Symposium (GIIS).

[12] C. Tapparello, H. Ayatollahi, W. Heinzelman "Energy Harvesting Framework for Network Simulator 3 (ns-3)", University of Rochester, Rochester, New York, November 2014.

[13] Pasca, S. T. V., Akilesh B., Anand A. V., Tamma B. R., "A NS-3 Module for LTE UE Energy Consumption", 2016 IEEE International Conference on Advanced Networks and Telecommunications Systems.

5th Workshop on Internet of Things—Enablers, Challenges and Applications

THE Internet of Things is a technology which is rapidly emerging the world. IoT applications include: smart city initiatives, wearable devices aimed to real-time health monitoring, smart homes and buildings, smart vehicles, environment monitoring, intelligent border protection, logistics support. The Internet of Things is a paradigm that assumes a pervasive presence in the environment of many smart things, including sensors, actuators, embedded systems and other similar devices. Widespread connectivity, getting cheaper smart devices and a great demand for data, testify to that the IoT will continue to grow by leaps and bounds. The business models of various industries are being redesigned on basis of the IoT paradigm. But the successful deployment of the IoT is conditioned by the progress in solving many problems. These issues are as the following:

- The integration of heterogeneous sensors and systems with different technologies taking account environmental constraints, and data confidentiality levels;
- Big challenges on information management for the applications of IoT in different fields (trustworthiness, provenance, privacy);
- Security challenges related to co-existence and interconnection of many IoT networks;
- Challenges related to reliability and dependability, especially when the IoT becomes the mission critical component;
- Zero-configuration or other convenient approaches to simplify the deployment and configuration of IoT and self-healing of IoT networks;
- Knowledge discovery, especially semantic and syntactical discovering of the information from data provided by IoT.

The IoT technical session is seeking original, high quality research papers related to such topics. The session will also solicit papers about current implementation efforts, research results, as well as position statements from industry and academia regarding applications of IoT. The focus areas will be, but not limited to, the challenges on networking and information management, security and ensuring privacy, logistics, situation awareness, and medical care.

TOPICS

The IoT session is seeking original, high quality research papers related to following topics:

- Future communication technologies (Future Internet; Wireless Sensor Networks; Web-services, 5G, 4G, LTE, LTE-Advanced; WLAN, WPAN; Small cell Networks…) for IoT,
- Intelligent Internet Communication,
- IoT Standards,
- Networking Technologies for IoT,
- Protocols and Algorithms for IoT,
- Self-Organization and Self-Healing of IoT Networks,
- Object Naming, Security and Privacy in the IoT Environment,
- Security Issues of IoT,
- Integration of Heterogeneous Networks, Sensors and Systems,
- Context Modeling, Reasoning and Context-aware Computing,
- Fault-Tolerant Networking for Content Dissemination,
- IoT Architecture Design, Interoperability and Technologies,
- Data or Power Management for IoT,
- Fog—Cloud Interactions and Enabling Protocols,
- Reliability and Dependability of mission critical IoT,
- Unmanned-Aerial-Vehicles (UAV) Platforms, Swarms and Networking,
- Data Analytics for IoT,
- Artificial Intelligence and IoT,
- Applications of IoT (Healthcare, Military, Logistics, Supply Chains, Agriculture, ...),
- E-commerce and IoT.

The session will also solicit papers about current implementation efforts, research results, as well as position statements from industry and academia regarding applications of IoT. Focus areas will be, but not limited to above mentioned topics.

TECHNICAL SESSION CHAIRS

- **Cao, Ning,** College of Information Engineering, Qingdao Binhai University
- **Chudzikiewicz, Jan,** Military University of Technology, Poland
- **Zieliński, Zbigniew,** Military University of Technology, Poland

PROGRAM COMMITTEE

- **Al-Anbuky, Adnan,** Auckland University of Technology, New Zealand
- **Antkiewicz, Ryszard,** Military University of Technology, Poland
- **Brida, Peter,** University of Zilina, Slovakia
- **Chudzikiewicz, Jan,** Military University of Technology in Warsaw, Poland

- **Cui, Huanqing,** Shandong University of Science and Technology, China
- **Ding, Jianrui,** Harbin Institute of Technology, China
- **Fouchal, Hacene,** University of Reims Champagne-Ardenne, France
- **Fuchs, Christoph,** Fraunhofer Institute for Communication, Information Processing and Ergonomics FKIE, Germany
- **Hodoň, Michal,** University of Žilina, Slovakia
- **Johnsen, Frank T.,** Norwegian Defence Research Establishment (FFI), Norway
- **Karpiš, Ondrej,** University of Žilina, Slovakia
- **Krco, Srdjan,** DunavNET
- **Laqua, Daniel,** Technische Universit"at Ilmenau, Germany
- **Lenk, Peter,** NATO Communications and Information Agency, Other
- **Li, Guofu,** University of Shanghai for Science and Technology, China
- **Marks, Michał,** NASK - Research and Academic Computer Network, Poland
- **MURAWSKI, Krzysztof,** Military University of Technology, Poland
- **Papaj, Jan,** Technical university of Košice, Slovakia
- **Savaglio, Claudio,** University of Calabria, Italy
- **Ševčík, Peter,** University of Žilina, Slovakia
- **Shaaban, Eman,** Ain-Shams university, Egypt
- **Staub, Thomas,** Data Fusion Research Center (DFRC) AG, Switzerland
- **Suri, Niranjan,** Institute of Human and Machine Cognition, United States
- **Wrona, Konrad,** NATO Communications and Information Agency

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 343–346
DOI: 10.15439/2021F24
ISSN 2300-5963 ACSIS, Vol. 25

UAV Mission Definition and Implementation for Visual Inspection

Tomáš Adam
Department of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Letná 9, 040 01 Košice, Slovakia
Email: tomas.adam@tuke.sk

František Babič
Department of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Letná 9, 040 01 Košice, Slovakia
Email: frantisek.babic@tuke.sk

Abstract— **Although Unmanned Aerial Vehicles (UAVs) are currently still in a state of continuous development, the trend of their integration into a wide range of industries and applications is growing. It leads to risk and cost reduction but requires skilled operators. This fact motivated us to propose a new approach towards a UAV-based flight mission-definition system based on state-of-the-art waypoint-based techniques. Our solution enables the configuration and autonomous conduction of flight trajectories, whose spatial complexity exceeds the visualization capabilities of currently available solutions. Two testing scenarios confirmed our expectations; the autonomous execution of the trajectory reduced the time required in all cases by almost a half while achieving the same output as a user-controlled manual flight. The proposed solution extends the possibilities of users in creating complex flight trajectories and significantly contributes to the higher time efficiency of recurrent flights.**

I. Introduction

WHILE the potential of the UAVs has been used in the past, mainly in the military sector, the continuing trend of increasing integration in the civil and commercial spheres has made it possible to increase the effectiveness of several activities in various fields. The potential use of these devices supports existing areas and helps to create new segments of application and development for commercial and civil purposes. The major benefit of UAVs excels mainly in the activities whose character requires, above all, unrestricted movement in space. This feature contributes to increasing the safety of the participating users and creates a tool that provides a view from different perspectives without having to overcome vertical obstacles.

A. Related Work

UAVs in agriculture bring several benefits, especially in the field of multi-spectral scanning and mapping. Vegetation indices of the infrared part of the light spectrum, for instance, the Normalized Difference Vegetation Index (NDVI) [1] can be used for the vegetation mapping, which can provide information about the vegetation stress, soil composition, fertilization quality, pest infestations, nutrient deficiencies, water stress, and other relevant conditions affecting crop productivity. This approach makes the capturing of induced abiotic stress before the reactions begin to appear in the visible part of the light spectrum [2].

UAVs in the energy industry find application in many scenarios, but mainly in control and prevention activities or damage investigations after natural disasters. Using UAVs in solar power plants allows regular inspection and maintenance of photovoltaic panels placed on inclined structures and inaccessible places through thermo-graphic imaging and diagnostics. This approach is currently the cheapest and fastest way to maintain and identify hot spots (which are among the defects that may cause the most destructive effects) in large solar power plants without the need to install fixed sensors and cameras [3, 4].

Checking the condition of wind power plants with UAVs allows access to wind turbines and their blades, which can currently reach a length of more than a hundred meters without using costly elevating platforms, scaffoldings, and industrial climbing methods [5]. According to Ian Glenn, CEO of ING Robotic Avionics, inspecting a single wind turbine using UAVs is 50% cheaper and takes an average of 1.5 to 3 hours, which is 3 to 4 times more time-efficient than conventional methods[1]. This process's availability and efficiency help increase the life of entire wind farms and prevent damage that, if neglected, may necessitate the replacement of the entire wind turbine blade [6].

The transmission and distribution networks are complex system with many components whose failure can lead to serious problems. During power line inspections, UAVs have significant advantages compared to conventional human expert-based and helicopter-based methods. They allow automated and contactless inspection of electric power distribution systems in high resolution with minimal downtimes and interruptions in the power supply [7]. Unlimited movement in space makes it possible to achieve higher efficiency of the visual inspection, allowing a point of view of individual components from several angles that cannot be reached during conventional inspections [8]

[1] http://insideunmannedsystems.com/ing-robotic-aviation-uses-responder-uas-for-turbine-blade-inspections/

The paper is organized as follows: the introduction briefly presents the motivation for proposing the new solution for visual inspections using UAVs. The second chapter describes the solution's design, including the user interface for flying missions' definition and implementation. The experiments cover two scenarios demonstrating the benefits of the proposed solution; the final chapter concludes the paper.

II. Proposed Solution

This section describes briefly the technological details of our solution. The architecture is composed of two primary subsystems (Fig. 1). The Ground Control Station (GCS) module and the Unmanned Aerial Vehicle (UAV).

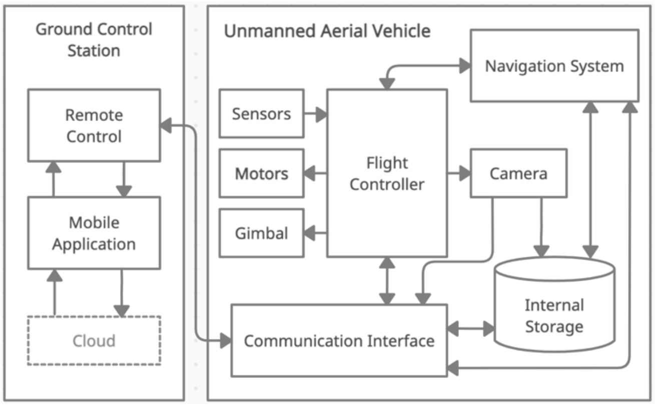


Fig. 1 Mission definition system

The UAV represents the quadrotor itself, which includes multiple cooperating parts: the flight controller (FC) performs mission execution and the flight based on the instructions from GCS. Then navigation system, helping with the orientation and correction of the actual UAV position with desired flight trajectory. The capture device (video, thermal camera, or lidar) performs the measurements and visual inspection. The captured data of these measurements should be stored in the internal SD card storage of the UAV with a time and GPS reference. The communication interface provides the primary wireless communication between UAV and GCS.

The Ground Control Station (GCS) is composed of several components. The remote controller represents the main component for wireless communication with the UAV, providing mainly user-based real-time flight instructions via joystick levers and buttons. It also provides communication access for the mobile application functions (as detailed in Section B) with a mission-definition interface and cloud storage services.

A. *Mission Visualization and Preparation*

The iOS mobile application represents the graphical user interface (GUI) for user interaction with specific functions, which exceeds the UAV remote controller capabilities. The primary function of this part aims at flight mission definition and its preparation for execution. The internal architecture depicted in Fig. 2 shows the flight mission specification conversion from data defined by the user to the mission flight plan used by the UAV and trajectory visualization in the GUI. These data may represent the previously saved flight mission, or the actual flight trajectory created by the user through GUI. The process can be decomposed into several phases.

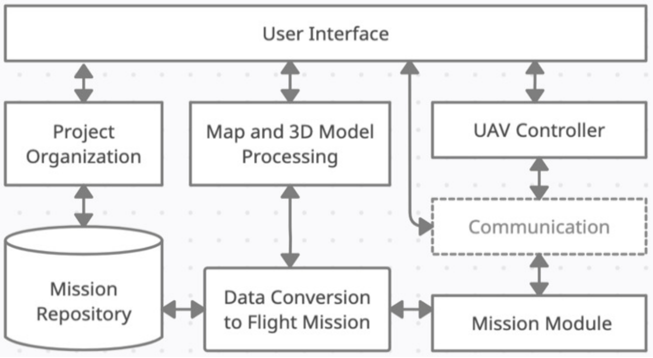
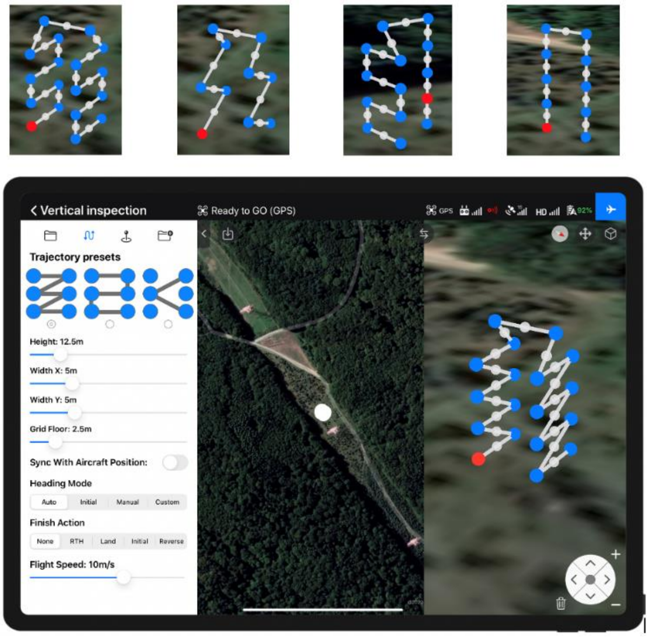


Fig. 2 Internal architecture

First, the mission specification data defined in the GUI is converted from the flight trajectory visualization module to a flight mission and saved into the database. Then the flight mission is validated and prepared for execution by the UAV. Finally, after validation, successful preflight control of the UAV peripherals, and loading the flight mission into its internal memory, the mission execution module starts the defined path execution.

B. *Mission Definition Interface*

The main layout depicted in Figure 3. comprises two main parts: mission definition and trajectory visualization. The mission definition part is represented by multiple sections containing saved flight missions, trajectory definition, and waypoint configuration. The parameters in this panel change according to the selected mission or waypoint with defined GPS coordinates, altitude, rotation, flight speed, and action.


Vertical inspection

Ready to GO (GPS)

Trajectory presets

Height: 12.5m

Width X: 5m

Width Y: 5m

Grid Floor: 2.5m

Sync With Aircraft Position:

Heading Mode: [✓ Auto] Initial Manual Custom

Finish Action: [✓ None] RTH Land Initial Reverse

Flight Speed: 10m/s

Fig. 3 Main interface of the mission definition tool

The second part is represented by a 2D and 3D trajectory visualization to provide the required level of customization for the user, without limiting the ability to edit and display the complex flight trajectories.

III. Experiments and Results

Two scenarios were chosen to demonstrate the benefits of the proposed solution and the increase of the UAV flight efficiency on basic flight trajectories.

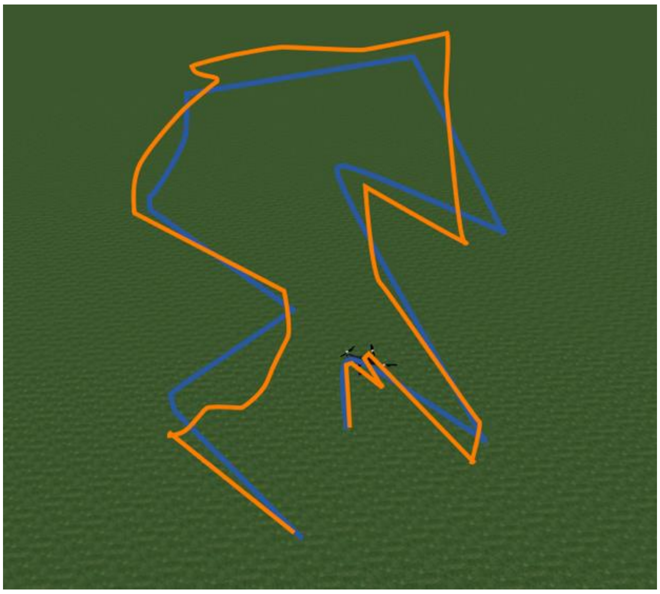


Fig. 4 Mission trajectory visualization

The first scenario illustrates the comparison of the autonomous flight of the UAV representing the ideal flight trajectory created through the mission definition interface, with the similar flight trajectory performed manually by the user. These trajectories are color-coded, where blue indicates an autonomous flight and orange a manual flight conducted by the user. The primary focus of these test flights was to compare GPS coordinates (Fig. 4, 5, and 6), the flight altitude, and the time required to complete the task.

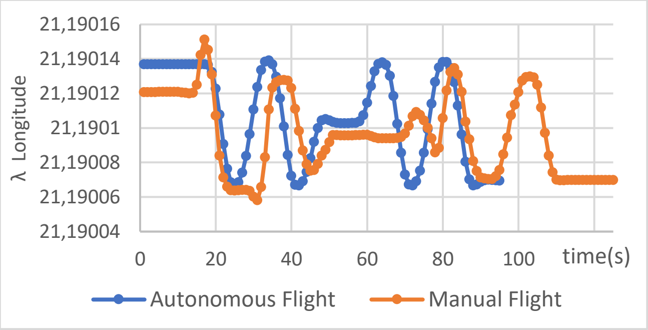


Fig. 5 Visualization of the first test: Longitude

For a more suitable display and simplification of recording deviations in the flight performance, a trajectory of 5x5x10 meters was chosen with a northeast wind of 5 m/s to increase the authenticity of the simulation. Ten flights were performed during the testing, from which the flight with the lowest deviation from the autonomously performed reference trajectory was selected (Fig. 4). During these flights, the trajectories were recorded and visualized through graphs showing the change in latitude and longitude values over time (Figs. 5 and 6).

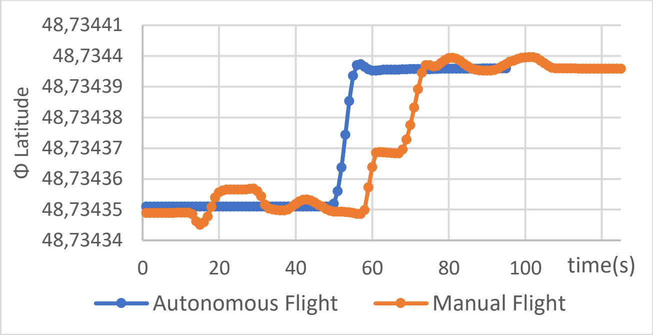


Fig. 6 Visualization of the first test: Latitude

Visualization of the overall course of the test (Fig. 5 and 6) shows that the manual execution of the trajectory took 35% longer and contained several errors caused mainly by the inertia of the UAV and the time required for rotation in the upper part of the flight trajectory.

The second scenario illustrates the time effectiveness of a repetition of the flight trajectory flown by the user manually. In this part, the main focus was on the ability of the proposed application to replicate the flight trajectory and the performed actions. The reference trajectory represents a simulation of the visual inspection of a large structure with the rotation of the device and the creation of images at different height levels.

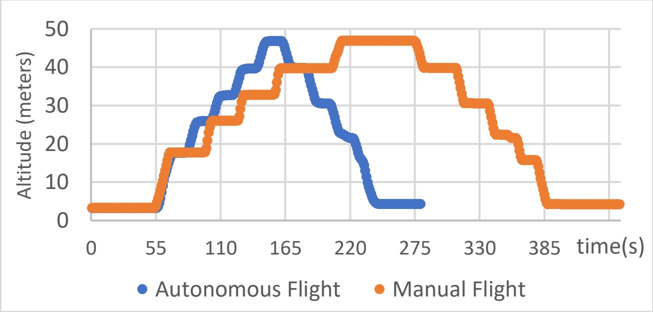


Fig. 7 Visualization of the second test: Altitude

Repeated execution of the trajectory autonomously took on average 45% less time (Fig. 7) than the user's original flight while maintaining the same rotation of the UAV, camera tilt and quality of the created images.

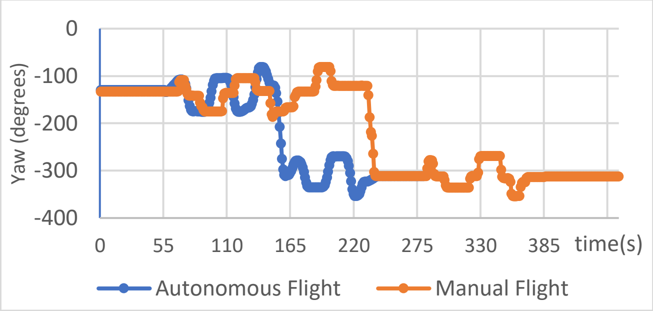


Fig. 8 Visualization of the second test: Yaw

This significant reduction in time was caused by a continuous change of the UAV's attitude and the camera tilt during the flight between the individual waypoints on the trajectory (Fig. 8). During the manual flight, this operation required stopping and adjusting the appropriate tilt of the camera manually.

The increase in the time efficiency of the flight performance is also visible in Figure 9, which visualizes the development of the GPS coordinates. The autonomous flight minimizes the required time to find an appropriate viewpoint at individual waypoints, to stabilize the UAV, focus, and record creation.

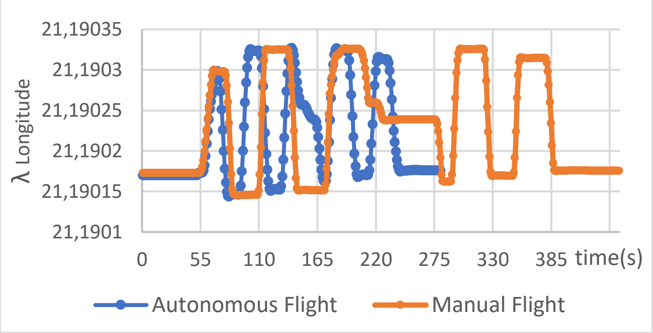


Fig. 9 Visualization of the second test: Longitude

The time efficiency in repeating the flight trajectory is much higher, as no emphasis was placed on the quality and complexity of the inspection process itself when creating the reference trajectory. For a more appropriate demonstration of the developed solution, the reference point for taking the images was the approximate center of the flight trajectory, which enabled the creation of several waypoints with different points of view in a relatively short time. When performing the actual inspection work by a professional, the time required to find an appropriate point of view is much higher and requires a more precise flight with a more extensive number of created images of better quality.

IV. Conclusion and Future Work

This paper presents an approach towards a UAV-based flight mission-definition system enabling the configuration and autonomous conduction of flight trajectories. The proposed solution was tested and verified in a simulated environment with two different types of testing, comparing the autonomous flight of the UAV with the manual flight performed by the user.

The efficiency comparison of flight trajectory execution shows that, in general, the performance is good enough for the intended purpose. During testing, the autonomous execution of the trajectory reduced the time required in all cases by almost half while achieving the same output as a user-controlled manual flight. The 3-dimensional flight trajectory visualization expands the orthographic view used in similar solutions and allows detailed configuration of the flight trajectory from multiple perspectives without limitation. The proposed solution not only extends the possibilities of users in creating complex flight trajectories but also significantly contributes to the higher time efficiency of recurrent flights. These recurrent flights allow less experienced pilots to perform routine work without significant time delays and the potential risk of damaging the UAV. Also, replacing the physical presence of workers in an inaccessible and potentially hazardous environment with machines allows the transfer of the value of highly qualified experts to a safer environment on the ground.

The current state of implementation represents only a part of the final solution and provides several possibilities for expansion in the future. With a web interface, the application can serve as an enterprise solution for managing objects of entire industrial infrastructures with multiple operators and UAVs. Implementing Artificial Intelligence like Machine Learning, Big Data and Computer Vision algorithms can significantly increase the possibilities of automation in post-processing and routine inspections, thanks to the analysis of the created records and the evaluation of the degradation of scanned objects over time.

Acknowledgment

The work was supported by The Slovak Research and Development Agency under grant no. APVV-16-0213.

References

[1] Xue, J., & Su, B. (2017). Significant remote sensing vegetation indices:A review of developments and applications. *Journal of Sensors*, *2017*. https://doi.org/10.1155/2017/1353691

[2] Radoglou-Grammatikis, P., Sarigiannidis, P., Lagkas, T., & Moscholios, I. (2020). A compilation of UAV applications for precision agriculture. *Computer Networks*, *172*(January), 107148. https://doi.org/10.1016/j.comnet.2020.107148

[3] Arenella, A.; Greco, A.; Saggese, A.; Vento, M., Real Time Fault Detection in Photovoltaic Cells by Cameras on Drones. In Image Analysis and Recognition, Proceedings ofthe 14th International Conference, ICIAR 2017, Montreal, QC, Canada, 5–7 July 2017; Karray, F., Campilho, A., Cheriet, F., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 617–625.

[4] Addabbo, P., Angrisano, A., Bernardi, M. L., Gagliarde, G., Mennella, A., Nisi, M., & Ullo, S. L. (2018). UAV system for photovoltaic plant inspection. *IEEE Aerospace and Electronic Systems Magazine*, *33*(8), 58–67. https://doi.org/10.1109/MAES.2018.170145

[5] Hallermann, N., & Morgenthal, G. (2013). Unmanned aerial vehicles (UAV) for the assessment of existing structures. *Long Span Bridges and Roofs - Development, Design and Implementation*, *September*. https://doi.org/10.2749/222137813808627172

[6] Stokkeland, M., Klausen, K., & Johansen, T. A. (2015). Autonomous visual navigation of Unmanned Aerial Vehicle for wind turbine inspection. *2015 International Conference on Unmanned Aircraft Systems, ICUAS 2015*, 998–1007. https://doi.org/10.1109/ICUAS.2015.7152389

[7] Tudevdagva, U., Battseren, B., Hardt, W., Blokzyl, S., & Lippmann, M. (2017). UAV-based Fully Automated Inspection System for High Voltage Transmission Lines Unmanned Aerial Vehicle-Based Fully Automated Inspection System for High Voltage Transmission Lines. *Automation and software engineering* , 1(19).

[8] Liu, X., Miao, X., Jiang, H., & Chen, J. (2020). Data analysis in visual power line inspection: An in-depth review of deep learning for component detection and fault diagnosis. *Annual Reviews in Control*, *50*(June), 253–277. https://doi.org/10.1016/j.arcontrol.2020.09.002

2$^{\text{nd}}$ International Forum on Cyber Security, Privacy and Trust

NOWADAYS, information security works as a backbone for protecting both user data and electronic transactions. Protecting communications and data infrastructures of an increasingly inter-connected world have become vital nowadays. Security has emerged as an important scientific discipline whose many multifaceted complexities deserve the attention and synergy of computer science, engineering, and information systems communities. Information security has some well-founded technical research directions which encompass access level (user authentication and authorization), protocol security, software security, and data cryptography. Moreover, some other emerging topics related to organizational security aspects have appeared beyond the long-standing research directions.

The International Forum of Cyber Security, Privacy, and Trust (NEMESIS'21) as a successor of International Conference on Cyber Security, Privacy, and Trust (INSERT'19) focuses on the diversity of the cyber information security developments and deployments in order to highlight the most recent challenges and report the most recent researches. The session is an umbrella for all cyber security technical aspects, user privacy techniques, and trust. In addition, it goes beyond the technicalities and covers some emerging topics like social and organizational security research directions. NEMESIS'21 serves as a forum of presentation of theoretical, applied research papers, case studies, implementation experiences as well as work-in-progress results in cyber security. NEMESIS'21 is intended to attract researchers and practitioners from academia and industry and provides an international discussion forum in order to share their experiences and their ideas concerning emerging aspects in information security met in different application domains. This opens doors for highlighting unknown research directions and tackling modern research challenges. The objectives of the NEMESIS'21 can be summarized as follows:

- To review and conclude research findings in cyber security and other security domains, focused on the protection of different kinds of assets and processes, and to identify approaches that may be useful in the application domains of information security.
- To find synergy between different approaches, allowing elaborating integrated security solutions, e.g. integrate different risk-based management systems.
- To exchange security-related knowledge and experience between experts to improve existing methods and tools and adopt them to new application areas

Topics

- Biometric technologies
- Cryptography and cryptanalysis
- Critical infrastructure protection
- Security of wireless sensor networks
- Hardware-oriented information security
- Organization- related information security
- Social engineering and human aspects in cyber security
- Individuals identification and privacy protection methods
- Pedagogical approaches for information security education
- Information security and business continuity management
- Tools supporting security management and development
- Decision support systems for information security
- Trust in emerging technologies and applications
- Digital right management and data protection
- Threats and countermeasures for cybercrimes
- Ethical challenges in user privacy and trust
- Cyber and physical security infrastructures
- Risk assessment and management
- Steganography and watermarking
- Digital forensics and crime science
- Security knowledge management
- Security of cyber-physical systems
- Privacy enhancing technologies
- Trust and reputation models
- Misuse and intrusion detection
- Data hide and watermarking
- Cloud and big data security
- Computer network security
- Assurance methods
- Security statistics

Technical Session Chairs

- **Awad, Ali Ismail,** Luleå University of Technology, Sweden
- **Bialas, Andrzej,** Research Network Łukasiewicz – Institute of Innovative Technologies EMAG, Poland

Program Committee

- **Banach, Richard,** University of Manchester, United Kingdom
- **Bun, Rostyslav,** Lviv Polytechnic National University, Ukraine
- **Clarke, Nathan,** Plymouth University, United Kingdom
- **Cyra, Lukasz,** DM/OICT/RMS (UN)

- **Daszczuk, Wiktor Bohdan,** Warsaw University of Technology, Poland
- **Felkner, Anna,** Research and Academic Computer Network NASK
- **Furnell, Steven,** Plymouth University, United Kingdom
- **Furtak, Janusz,** Military University of Technology, Poland
- **Gawkowski, Piotr,** Institute of Computer Science, Warsaw University of Technology, Poland
- **Grzenda, Maciej,** Orange Labs Poland and Warsaw University of Technology, Poland
- **Hämmerli, Bernhard M.,** Hochschule für Technik+Architektur (HTA), Switzerland
- **Hasssaballah, M.,** South Valley University, Egypt
- **Kapczynski, Adrian,** Silesian University of Technology, Poland
- **Krendelev, Sergey,** Novosibirsk State University, JetBrains research, Russia
- **MD Faisal, Mohammad,** Integral University, India
- **MD Rafiqul, Islam,** School of Computing and Mathematics|Charles Sturt University
- **Misztal, Michal,** Military University of Technology, Poland
- **Pańkowska, Małgorzata,** University of Economics in Katowice, Poland
- **Rot, Artur,** Wroclaw University of Economics, Poland
- **Stokłosa, Janusz,** WSB University in Poznan, Poland
- **Suski, Zbigniew,** Military University of Technology, Poland
- **Szmit, Maciej,** University of Lodz, Poland
- **Wahid, Khan Ferdous,** Airbus, Germany
- **Yahya, Eslam,** Ohio State University, Columbus
- **Zamojski, Wojciech,** Wrocław University of Technology
- **Zieliński, Zbigniew,** Military University of Technology, Poland

DOI: 10.15439/2021F85
ISSN 2300-5963 ACSIS, Vol. 25

Identification of Unintentional Perpetrator Attack Vectors using Simulation Games: A Case Study

Martin Macak, Stefan Bojnak, Barbora Buhnova
Faculty of Informatics, Masaryk University
Brno, Czech Republic
{macak, bojnak, buhnova}@mail.muni.cz

***Abstract*—In our digital era, insider attacks are among the serious underresearched areas of the cybersecurity landscape. A significant type of insider attack is facilitated by employees without malicious intent. They are called unintentional perpetrators. We proposed mitigating these threats using a simulation-game platform to detect the potential attack vectors. This paper introduces and implements a scenario that demonstrates the usability of this approach in a case study. This work also helps to understand players' behavior when they are not told upfront that they will be a target of social engineering attacks. Furthermore, we provide relevant acquired observations for future research.**

I. Introduction

INSIDER attacks are one of the most significant cybersecurity challenges, as they are more difficult to detect than external attacks since insiders are employees with authorized access to the organization's resources [1]. They can be seen as a complicated process that consists of multiple steps [2]. The most commonly recognized insiders are malicious insiders who know the organization and can act inconspicuously [3]. However, insider attacks do not have to be caused by malicious intent. They can be caused or facilitated by so-called unintentional perpetrators, complicating their detection even more [4].

The prevention of unintentional perpetrator attacks in organizations is very important [5]. We suggest advancing the research by creating a game-based cybersecurity training platform to identify the unintentional perpetrator attack vectors [5]. It is able to provide complex simulation games that combine an environment for both human-based and computer-based social engineering attacks. It also logs each relevant activity, providing data for the process analysis of players' playthrough, from which we can get possible attack vectors that can be enabled in the future in a real non-simulated scenario in an organization.

This work performs an initial case study on the aforementioned platform to understand the behavior of participants who played the game. We specifically inspect their behavior in the situation when they are not told upfront that they will be targeted by several social engineering attacks. Furthermore, we provide a set of other observations from this case study, e.g., players' perception of the attacks and their reactions to them. It provides valuable information for future work for the researchers in the area of insider attack prevention by cybersecurity training.

The remaining of the paper is structured as follows. Section II provides the relevant related work to our platform, which is subsequently described in Section III. In Section IV, we specify the designed game scenario, which is then evaluated in Section V. Furthermore, Section VI provides the threats to validity of this work. Afterward, Section VII concludes the paper.

II. Related Work

Unintentional perpetrator threat research can be divided into two areas: behaviorally-focused and technically-focused [5].

Liu et al. [4] performed a technically-oriented survey targeting both malicious and unintentional insider attacks. They provide a review of detection and prevention techniques. One of their main points is that the prevention of insider attacks is generally less considered than their detection. It was also confirmed in a similar survey by Homoliak et al. [6], which added, among others, that the trend of unintentional insider threats and attacks is increasing.

The prevention techniques like the deployment of authentication techniques [4], access control [7], least privileges, information security policy [8], firewall, antivirus, and encryption [9] are beneficial. However, their effectiveness is influenced by the human factor in the organization.

The human factor in cybersecurity is frequently studied, for example, sources of stress related to compliance with security policies [10]. Another study [11] shows that some users tend to believe that security technology will protect them, regardless of their behavior. This leads to negligence and severe security vulnerability.

Malicious attackers can directly exploit these security vulnerabilities. Social engineering techniques are commonly used for this task and are considered very dangerous, as they cannot be mitigated by technology alone [12].

Cybersecurity training platforms can be considered as a tool for addressing this issue. Their development has seen a massive increase in recent years [13]. Currently, there is plenty of various cyber ranges that emulate computer networks and then support the organization of hands-on cybersecurity training, e.g., KYPO Cyber Range [14], Michigan Cyber Range [15], SimSpace Cyber Range [16], EDURange [17], DETERlab[18], CyRIS [19], or CyTrONE [20]. Their benefit is letting the participants from organizations experience the attacks, either from the attacker's or the defender's view, thus getting equipped to mitigate the attacks in the future.

The main limitation of the cyber ranges is that they can be hard to organize and are often not focused on non-IT experts [5]. Furthermore, in some cases, even if participants had several hours of practice, the majority of them still clicked on a phishing link after training fulfillment [21]. Therefore, security response efficacy is an important aspect of security training because employees have to be convinced that information and recommendations gained during security education are reliable, practical, and functional. This can be achieved via a game. Gamification elements increase the overall enjoyment during learning, which is essential for training efficiency [22].

In our work, we focus on people who are not cybersecurity experts, and we bring the training to them instead of bringing them to the training. We are gamifying the training and studying their perception of it when we do not specifically mention that it has a cybersecurity purpose.

III. Platform Description

In this section, we briefly describe the general overview of the platform, game application, analysis technique – process mining, and the structure of the captured event logs.

A. General overview

A primary requirement for the platform is to have two separate applications for two main actors. The first application is a game for players – these are potentials victims of social engineering attacks in organizations. The second is a scenario maker, where a game designer can design scenarios for the game. A game designer is a person that represents an organization that tries to find potential insider threats among the behavior of its employees.

Figure 1 reflects basic game requirements as use cases of the player and game-designer actors:

- **Choose different games.** The game will be able to allow the player to play different scenarios.
- **Play a game**. The player will be able to play the scenario from the beginning to the end.

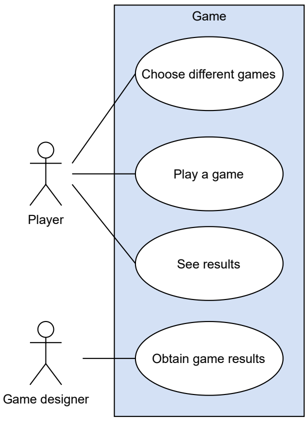


Fig. 1: Game use cases

- **See results**. The player will be able to see their own results of the game.
- **Obtain game results.** The game designer will be able to obtain the results of the game for each player. These results must reflect the whole player's flow and their decisions in the game.

Figure 2 shows requirements displayed as game designer use cases for the scenario-maker application. The requirements cover basic operations with a scenario:

- **Create scenario.** It allows a game designer to create a new scenario that they can import into a game.
- **Update scenario.** The game designer can update a previously created scenario.
- **Delete scenario.** The use case allows the removal of a previously created scenario.

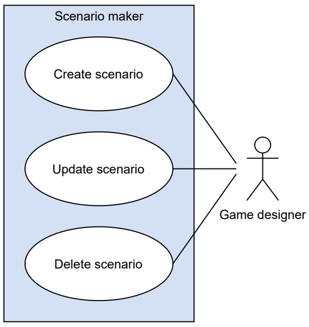


Fig. 2: Scenario maker use cases

B. Game application

The game is designed as a simulator of the insiders' workflow, so it looks like a simplified graphical user interface of an operating system. It consists of several modules that have their user interface. In this work, we used the following modules:

- **Desktop**. Represents the default module.
- **Email client**. Allows the player to receive and send emails. It is the main communication channel in the PC environment.
- **Web browser**. Allows the player to perform tasks received via email. Creates the fun element in the game and simulates the actual work.
- **Intermission**. A special module that consists of text that guides the players through the events that happen outside of the simulated system. For example, the events before arriving at work, coffee breaks, or group meetings. It provides the players the information about what actions are happening and can give them several options to react to these actions.

The reactions inside and outside the system are very important because of their possible impact in the future. Reactions outside the system are the options that players chose from *Intermission* module, like plugging the unknown flash drive into the computer or giving the access card to a stranger. Reactions inside the system might be actions like sending personal information to someone or reporting a phishing email.

C. *Process Mining*

Process mining techniques have proven to be very successful in 1) *process discovery*, which aims to find a descriptive model of the underlying process from event logs, 2) *conformance checking*, i.e., monitoring and inspecting whether the real execution of the process conforms to the corresponding designed (or discovered) reference process model, and 3) *process enhancement*, which improves and enriches a process model based on the related event data [23].

Process discovery is able to find a model that represents the process described in the event log. This model has to conform to four quality criteria – fitness, precision, generalization, and simplicity [23]. The model has low fitness when it can replay only a small number of traces in the event log. When the model has poor precision, it means that it allows a very different behavior from the behavior in the log. On the other hand, a model with low generalization allows only the behavior that was in the log. The simplicity of the model is connected to whether the model explains the behavior with the minimum necessary information. Process discovery has been first discussed in [24], which describes discovery methods in the context of software engineering processes. Similar to some later published techniques [25], [26], it was limited to sequential processes. One of the first discovery algorithms that handled the concurrency of events is the Alpha algorithm [27]. It produces a marked Petri net from an event log. Later, many other algorithms emerged, like variants of the Alpha algorithm [28], Heuristic Miner [29], Fuzzy miner [30], and DecMiner [31].

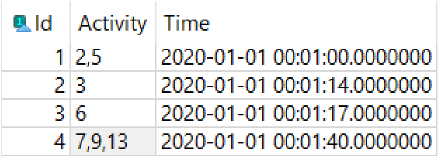
The purpose of conformance checking is to decide whether the execution of the process conforms to the corresponding process model [32]. Early conformance-checking techniques used token-based replay to detect non-fitting cases. They replayed a trace of events in a Petri net, and based on it, produced diagnostics [23]. For example, Conformance Checker [33] introduced two metrics: fitness and appropriateness. Fitness measures the degree to which the process model can replay the traces from the log. Appropriateness measures the simplicity, precision, and generalization of the model. However, the token-based approach often does not provide satisfactory results, so other alternatives, like alignment-based solutions, were introduced [34].

Process enhancement techniques aim to improve or extend an existing process model using information extracted from the process described in an event log [23]. This is important when the model does not reflect reality accurately. An example of process improvement is [35], where the authors repair the given model, increasing its fitness with respect to the given event log. In process extension, a new perspective is added to the process model, such as an organizational or time perspective. The approach in [36] uses the organizational perspective to enhance the model by roles of the activity originators. On the other hand, in [37], the time perspective is used.

D. *Data model of logs*

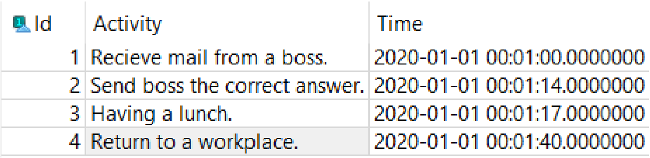
The unintentional perpetrator platform detection's primary purpose is to find possible threats created by a series of unintentionally wrong decisions of organization insiders by analyzing their behavior in a simulated environment. Logs are necessary for that use case because they record such behavior. In this case, analysis is done by process mining discovery, so logs must satisfy process mining conditions for event log data.

Our log data is stored in the database in a single independent table called Log. It has three columns: Id, Activity, and Timestamp. Id serves only as a primary key; timestamp represents the time from the beginning of the game. It means that if some activity is logged one minute after the game started, the timestamp will contain the value '2020-01-01 00:01:00.0000000'. The date is not essential and is set to 2020-01-01 and can be changed manually in the source code. The activity column records the component that the Player clicks with a combination of other checked components. By default, the application saves ids of components split by a comma into the Activity column, which can be changed in configuration to a string that better captures the activity's meaning. Figures 3 and 4 compare these two approaches with examples of the same logs, the first one with activity names through component ids, the second one through concise titles.



Id	Activity	Time
1	2,5	2020-01-01 00:01:00.0000000
2	3	2020-01-01 00:01:14.0000000
3	6	2020-01-01 00:01:17.0000000
4	7,9,13	2020-01-01 00:01:40.0000000

Fig. 3: Event logs without configured activities names



Id	Activity	Time
1	Recieve mail from a boss.	2020-01-01 00:01:00.0000000
2	Send boss the correct answer.	2020-01-01 00:01:14.0000000
3	Having a lunch.	2020-01-01 00:01:17.0000000
4	Return to a workplace.	2020-01-01 00:01:40.0000000

Fig. 4: Event logs with configured activities names

IV. Designed Scenario

In this section, we describe a scenario that we created for a case study of this platform. We created it in a web scenario maker, imported it to a game database, and tested it with multiple players.

A. *Description*

The scenario has to reflect the purpose of the platform – to detect possible insider threats in an organization. Our scenario should contain attack simulations of real social engineering attacks. The scenario should allow players to make decisions that lead to an attack or prevent an attack from happening.

A player in our scenario plays the game as an administrative employee of the MadeUp Ltd. company. The whole storyline is situated in one workday of this employee, starting when they come to work and finishing when they leave. During the game, the employee meets multiple tasks. Some of them are valid tasks assigned by the employee's bosses; an attacker has assigned others. A full story with all possibilities is displayed in the diagram that is in the Appendix.

B. *List of attacks*

The scenario contains four attempts of an attack on a player:

- **Card copy**. The attacker impersonates a building manager and tries to persuade the player that the player should lend them their access card so that the building manager can update it. The player gets the chance to refuse and report the attacker or give the attacker their card.
- **Phishing**. The attacker impersonates the boss and sends an email to the player with the information that there is a new employee in the company, and the player is the only one who has access to the accounting database. The attacker encourages the player that the player should send them this data back to that email. The player can send this data or refuse.
- **Flash drive**. The attacker pretends to be a new employee who received a flash drive with instructions about the company's internal system. The attacker tells the player that they do not understand these instructions and whether they would take the flash drive, read the instructions, and help them. The player has an opportunity to refuse to take the flash drive or not to plug it into their computer – having multiple opportunities to stop the attack.
- **Another phishing**. This phishing mail has a similar concept to the previous phishing attack, but the attacker develops more pressure on the player. The attacker pretends to be a company accountant and says that the player forgot to send the monthly report, so accounting cannot process their salary. However, if the player sends back the company number, accounting can fulfill the report on their behalf, and the player will get the salary.

V. Evaluation

In this section, we describe the data collected from the participants' testing. Firstly, we evaluate the questionnaire that was filled after the game completion. Then we discuss the event logs about participants' behavior in the game.

A. *Participants*

The group of participants was collected via social media on a voluntary basis. First, we acquired mostly students, so then we extended the invitation to cover also some participants who graduated already. We also aimed for a similar proportion of people working in the IT sector and those that do not. We aimed for at least 20 participants to acquire the appropriate amount of feedback and data about the behavior of this initial case study, which will help researchers in the future to design follow-up cybersecurity training and more advanced case studies.

B. *Questionnaire*

After each respondent had finished the game, they filled a survey that was designed with the purpose to answer the following two research questions:

1) Do the participants realize that they are targeted by a social engineering attack in our simulated environment?
2) Do the participants believe that they have behaved correctly during the game?

The survey also contains demographic questions to see what types of users tested the scenario and the game application. Overall, 25% of tested participants were women, 55% studied or worked in an IT-related area, 25% were students. The average age was 23, and the median age was 22. Half of the participants' highest education attained was high school, 20% had a bachelor's degree, and the rest of them had a master's degree.

We were interested in the overall impression of the game. The players had to choose a number from 1 to 5, where 1 means the best impression and 5 the worst. No player chose number 4 or 5, and the average impression is 1.95, so we evaluate players' overall image of the game as very good with some minor objections.

Further, we asked players whether they knew what to do during playing. The question checked the ergonomics of the scenario with the same five-point scale as in the previous question. It is essential to make players' user experience as comfortable as possible to focus on decision making and playing the simulation, not looking for what to do next. Answers give us an average of 2.25, which we judge as overall good. Some of the players gave feedback that they were unsure immediately what to do after playing a browser mini-game. Some of them did not know how to answer the mail with provided text paragraphs.

Participants further answered four questions in the questionnaire so that we could evaluate previously mentioned research questions. Figures 5, 6, 7, and 8 show these questions and their answers.

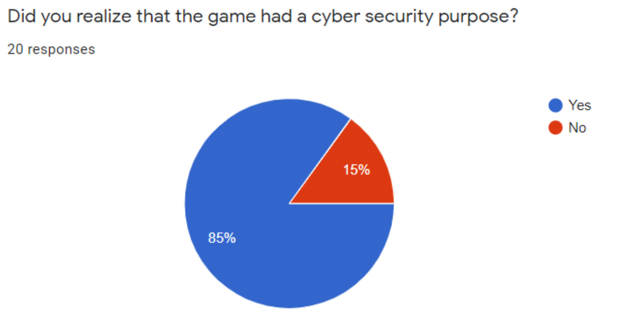


Fig. 5: Answers to the first question

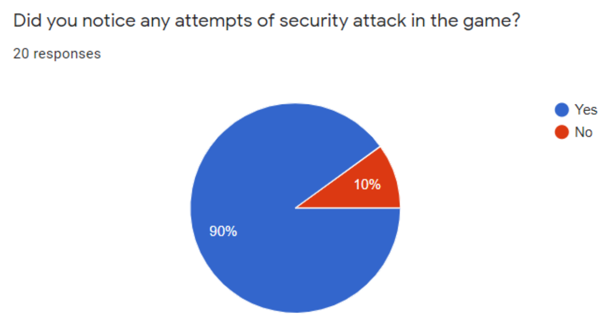


Fig. 6: Answers to the second question

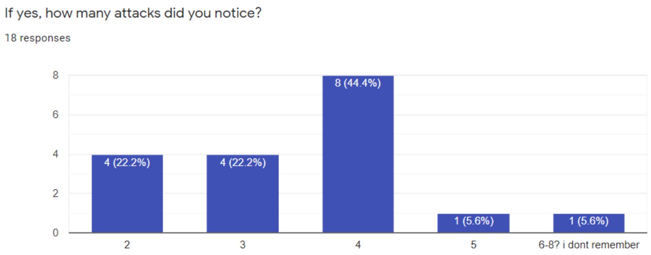


Fig. 7: Answers to the third question

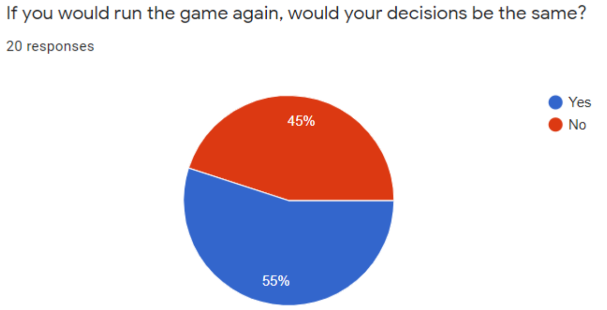


Fig. 8: Answers to the fourth question

The first three questions were linked to the first research question – how many people realized that they were targeted by a simulated attack. Most of the players observed that the game had a cybersecurity purpose. Many participants (90%) noticed that there were some attempts of attacks. There are four attacks in the scenario. Eight people noticed all of them, two people even more. Generally, from our testing sample, most people realized that they were victims of an attack in the simulation scenario, but many people still missed some of the attacks, in our case, 44.4% of them. When it comes to unintentional perpetrator attacks, such a percentage can be dangerous, and it may have significant unpleasant consequences – even one attack can lead to loss of clients' data, leak of personal or access information, or others. On the other hand, a few people noticed more attacks than there were actually present. This might be connected to the fact that when they realized the game had a cybersecurity purpose, they were much more careful.

The fourth question of the survey answers the second research question – how confident people were about their behavior after the security incident happened. Only 55% answered that their decision would be different in another simulation run in the same scenario. It means that only about half of test users were satisfied with their behavior. Therefore, we can assume that the game had a positive impact on their future behavior regarding cybersecurity.

C. Event logs

After the game was played, we obtained the event log from the game for analysis. Out of 20 participants, we were able to extract 19 cases. From this event log, we discovered a process model using Disco. Using this process mining approach, we were able to look more deeply into the process of players' playthrough and identify possible attack vectors. The model is in the Appendix in Figure 11.

We can see that the structure of activities is similar to the scenario story diagram. The thickness of the lines between activities tells how often players went this way in the game decision tree. The thicker the line, the more often they chose this particular path. Using Disco's interactive analysis, we can also see how long the players stayed in some activity and whether the players that became victims of some attack also became victims of another attack.

There are four activities in the event log of the prototype scenario representing successful attacks, which we mentioned earlier in this section. The process diagram shows whether someone made decisions that led to the incident and how many people have risen to the bait.

TABLE I: Success of simulated attacks

Attack	Success cases	Failed cases	Success rate
Card copy	4	15	21.05%
Fake boss phishing email	3	16	15.79%
Malicious flash drive	8	11	42.11%
Fake accounting phishing email	5	14	26.32%

Table I shows how many times the attacker was successful in each attack. The attack success rate was calculated as *(success cases / all cases) * 100*. The first attack was a card-copy attack. Four players gave an attacker their access card and let the attacker steal the card data. Another attack was the phishing attack with a fake boss. Three people sent accounting database access information to the attacker. The third attack involved a malicious flash drive from a false new employee. Eleven people took the flash drive, and eight of them plugged it into their personal computers. The last attack was also a phishing mail from a fake accounting department, leaving five successful incident cases.

Figure 9 demonstrates a part of the discovered process. This part shows the first attack from the prototype scenario. We can see there that each player except one stays when a random person stops them. Sixteen of them still stay when this person pretends to be a building maintainer, but twelve of them refuse to give them their access card. All of them meet in the activity

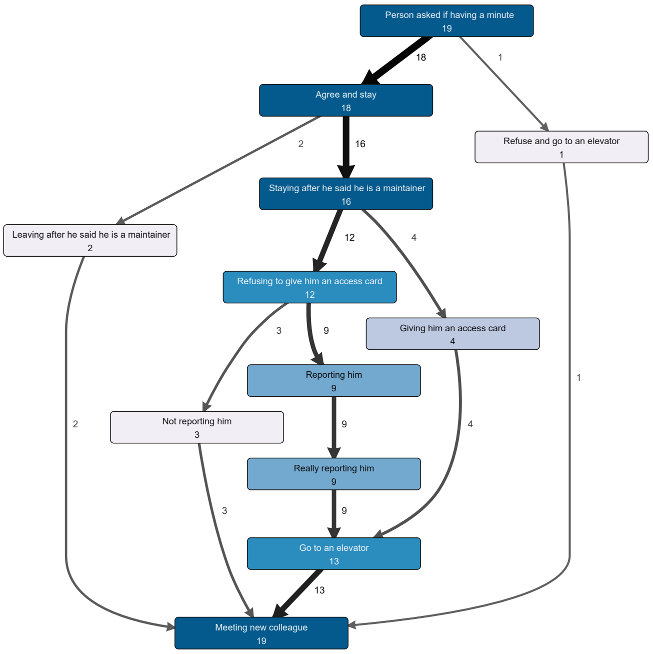

Person asked if having a minute
19
18
1
Agree and stay
18
Refuse and go to an elevator
1
2
16
Staying after he said he is a maintainer
16
Leaving after he said he is a maintainer
2
12
4
Refusing to give him an access card
12
3
9
Giving him an access card
4
Reporting him
9
1
2
Not reporting him
3
9
4
Really reporting him
9
3
9
Go to an elevator
13
13
Meeting new colleague
19

Fig. 9: Process diagram of the first attack

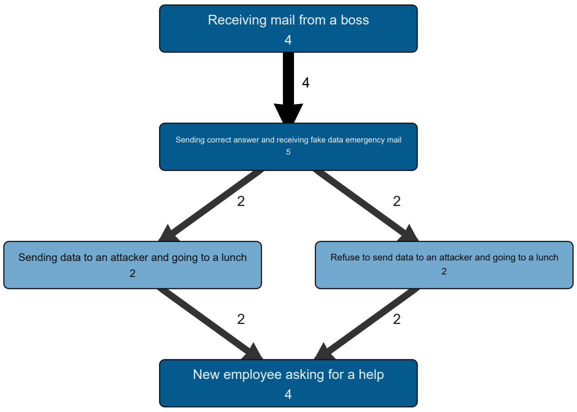

Receiving mail from a boss
4
4
Sending correct answer and receiving fake data emergency mail
5
2
2
Sending data to an attacker and going to a lunch
2
Refuse to send data to an attacker and going to a lunch
2
2
2
New employee asking for a help
4

Fig. 10: Process diagram of the second attack only with cases of the successful first attack

’Meeting new colleague’ because, after the attack attempt, the scenario sends everyone there

As an example of process analysis, we take four cases of a successful first attack (Activity ’Giving him an access card’ in Figure 9) and examine whether they also became victims of the phishing mail attack from a fake boss. Figure 10 shows that two cases send data to an attacker, and two players refuse. Similarly, organizations can analyze and detect where are the main gaps in their employees’ security knowledge.

The process diagram confirms survey results about the successful attacks with exact data of players’ behavior. As we see in Table I, each attack was successful, at least with some users, flash drive attack even in more than 40% of cases. It means that at least eight players out of 19 became victims of at least one attack – even when 85% of players realized that the game has a cybersecurity purpose. From our point of view, this number sounds alarming. It confirms that the prevention of unintentional perpetrator threats has to take a relevant place in the organizations’ cybersecurity prevention practice because not all users can observe and prevent social engineering attacks.

Overall, process mining helps us to better understand the behavior of players in the game. It generates the process model, which shows how exactly the players performed in the game. It shows not only the paths they followed but also the frequency of each action and the transition between the actions. Furthermore, we can analyze the game from the performance point of view and see how long each part took for the players. Utilizing it, we can find the problematic parts, identifying possible attack vectors via unintentional perpetrators in an organization.

VI. Threats to Validity

This section discusses the construct, internal, external, and conclusion validity of our work and threats to this validity.

A. *Construct validity threats*

We have carefully designed the game scenario to reflect the real situations that can happen to obtain the closest reactions we can. However, we are aware of the fact that many more situations can be employed, and we encourage the researcher community in the future to investigate the most effective scenarios.

B. *Internal validity threats*

We are aware that the confidence of usability of this platform might be biased because of the low number of participants. However, we aimed for a sufficient variability in participants for the current phase of our results. Therefore, we believe we provided interesting, relevant results that can help the community to take over from there. In the future, we encourage more case studies in organizations with multiple types of employees and even bigger variability.

C. *External validity threats*

It would be too early to generalize the results of this work beyond this case study. However, we have demonstrated that such a case study is possible and gathered essential aspects for future case studies.

D. *Conclusion validity threats*

We are aware that the current number of participants is not high enough to draw general conclusions. On the other hand, we believe that the value of this work is not primarily in the providing of general conclusions but in the reporting of the basic behavior of people in such types of games for the design of better future research studies.

VII. Conclusion

In this work, we performed an initial case study of a simulation game to identify the potential unintentional perpetrator attack vectors. We described the designed scenario and evaluated it on 20 participants to demonstrate its usefulness. It helped us understand the behavior of respondents who played the game and provided us with a set of relevant observations for future work in the research of cybersecurity training towards the prevention of insider attacks in organizations.

Future directions can be taken from multiple angles. More scenarios can be investigated further to get the guidelines for the effective scenarios. Moreover, more and bigger case studies with multiple types of employees in organizations will provide interesting results that can be generalized. Furthermore, in the future, we plan to utilize our own process mining application to incorporate more advanced process mining features that are not available in the Disco tool, like conformance checking, to provide much more detailed analysis results for potential unintentional perpetrator attack vectors, e.g., providing automatic hints for the analyst with interesting parts of the model.

Acknowledgment

This research was supported by ERDF "CyberSecurity, CyberCrime and Critical Information Infrastructures Center of Excellence" (No. CZ.02.1.01/0.0/0.0/16_019/0000822).

References

[1] J. Hong, J. Kim, and J. Cho, "The trend of the security research for the insider cyber threat," in *Security Technology*. Springer Berlin Heidelberg, 2009, pp. 100–107.

[2] M. Macak, I. Vanát, M. Merjavý, T. Jevočin, and B. Buhnova, "Towards process mining utilization in insider threat detection from audit logs," in *2020 Seventh International Conference on Social Networks Analysis, Management and Security (SNAMS)*, 2020, pp. 1–6.

[3] I. A. Gheyas and A. E. Abdallah, "Detection and prediction of insider threats to cyber security: a systematic literature review and meta-analysis," *Big Data Analytics*, vol. 1, no. 1, p. 6, 2016.

[4] L. Liu, O. De Vel, Q.-L. Han, J. Zhang, and Y. Xiang, "Detecting and preventing cyber insider threats: A survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 1397–1417, 2018.

[5] M. Macak, A. Kruzikova, L. Daubner, and B. Buhnova, "Simulation games platform for unintentional perpetrator attack vector identification," in *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*. ACM, 2020, p. 222–229.

[6] I. Homoliak, F. Toffalini, J. Guarnizo, Y. Elovici, and M. Ochoa, "Insight into insiders and it: A survey of insider threat taxonomies, analysis, modeling, and countermeasures," *ACM Comput. Surv.*, vol. 52, no. 2, Apr. 2019. [Online]. Available: https://doi.org/10.1145/3303771

[7] S. Sinclair and S. W. Smith, "Preventative directions for insider threat mitigation via access control," in *Insider Attack and Cyber Security*. Springer, 2008, pp. 165–194.

[8] T. Shimeall and R. Trzeciak, "Common sense guide to prevention and detection of insider threats," 01 2008.

[9] L. Cheng, F. Liu, and D. Yao, "Enterprise data breach: causes, challenges, prevention, and future directions," *WIREs: Data Mining and Knowledge Discovery*, vol. 7, no. 5, p. e1211, 2017.

[10] J. D'Arcy and P.-L. Teh, "Predicting employee information security policy compliance on a daily basis: The interplay of security-related stress, emotions, and neutralization," *Information & Management*, vol. 56, no. 7, p. 103151, 2019.

[11] T. Stafford, G. Deitz, and Y. Li, "The role of internal audit and user training in information security policy compliance," *Managerial Auditing Journal*, vol. 33, no. 4, pp. 410–424, 2018.

[12] F. Salahdine and N. Kaabouch, "Social engineering attacks: A survey," *Future Internet*, vol. 11, no. 4, p. 89, 2019.

[13] J. Davis and S. Magrath, "A survey of cyber ranges and testbeds," DTIC Document, Tech. Rep., 2013.

[14] J. Vykopal, R. Oslejsek, P. Celeda, M. Vizvary, and D. Tovarnak, "Kypo cyber range: Design and use cases," in *Proceedings of the 12th International Conference on Software Technologies - Volume 1: ICSOFT,*, INSTICC. SciTePress, 2017, pp. 310–321.

[15] MCR, "The Michigan Cyber Range." [Online]. Available: https://www.merit.edu/cyberrange/

[16] L. Rossey, "SimSpace cyber range," aCSAC 2015 Panel: Cyber Experimentation of the Future (CEF): Catalyzing a New Generation of Experimental Cybersecurity Research.

[17] R. Weiss, F. Turbak, J. Mache, and M. E. Locasto, "Cybersecurity education and assessment in edurange," *IEEE Security & Privacy*, no. 3, pp. 90–95, 2017.

[18] J. Mirkovic, T. V. Benzel, T. Faber, R. Braden, J. T. Wroclawski, and S. Schwab, "The Deter Project," 2010.

[19] C. Pham, D. Tang, K.-i. Chinen, and R. Beuran, "Cyris: A cyber range instantiation system for facilitating security training," in *Proceedings of the Seventh Symposium on Information and Communication Technology*, ser. SoICT '16. New York, NY, USA: ACM, 2016, pp. 251–258.

[20] R. Beuran, D. Tang, C. Pham, K.-i. Chinen, Y. Tan, and Y. Shinoda, "Integrated framework for hands-on cybersecurity training: CyTrONE," *Computers & Security*, vol. 78, pp. 43–59, 2018.

[21] A. J. Ferguson, "Fostering e-mail security awareness: The west point carronade," *Educause Quarterly*, vol. 28, no. 1, pp. 54–57, 2005.

[22] M. Silic and P. B. Lowry, "Using design-science based gamification to improve organizational security training and compliance," *Journal of Management Information Systems (JMIS)(accepted 01-Aug-2019)*, 2019.

[23] W. van der Aalst, *Process Mining: Data Science in Action*, 2nd ed. Springer Publishing Company, Incorporated, 2016.

[24] J. E. Cook and A. L. Wolf, "Automating process discovery through event-data analysis," in *Proceedings of the 17th International Conference on Software Engineering*, ser. ICSE '95. New York, NY, USA: Association for Computing Machinery, 1995, p. 73–82.

[25] A. Datta, "Automating the discovery of as-is business process models: Probabilistic and algorithmic approaches," *Information Systems Research*, vol. 9, no. 3, pp. 275–301, 1998.

[26] R. Agrawal, D. Gunopulos, and F. Leymann, "Mining process models from workflow logs," in *Advances in Database Technology — EDBT'98*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 467–483.

[27] W. van der Aalst, T. Weijters, and L. Maruster, "Workflow mining: Discovering process models from event logs," *IEEE transactions on knowledge and data engineering*, vol. 16, no. 9, pp. 1128–1142, 2004.

[28] B. F. van Dongen, A. A. De Medeiros, and L. Wen, "Process mining: Overview and outlook of petri net discovery algorithms," in *transactions on petri nets and other models of concurrency II*. Springer, 2009, pp. 225–242.

[29] A. Weijters, W. M. van der Aalst, and A. A. De Medeiros, "Process mining with the heuristics miner-algorithm," *Technische Universiteit Eindhoven, Tech. Rep. WP*, vol. 166, pp. 1–34, 2006.

[30] C. W. Günther and W. M. P. van der Aalst, "Fuzzy mining – adaptive process simplification based on multi-perspective metrics," in *Business Process Management*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 328–343.

[31] E. Lamma, P. Mello, M. Montali, F. Riguzzi, and S. Storari, "Inducing declarative logic-based models from labeled traces," in *Business Process Management*. Springer Berlin Heidelberg, 2007, pp. 344–359.

[32] J. Carmona, B. van Dongen, A. Solti, and M. Weidlich, *Conformance Checking*. Springer, 2018.

[33] A. Rozinat and W. M. van der Aalst, "Conformance checking of processes based on monitoring real behavior," *Information Systems*, vol. 33, no. 1, pp. 64–95, 2008.

[34] W. van der Aalst, A. Adriansyah, and B. van Dongen, "Replaying history on process models for conformance checking and performance analysis," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 182–192, 2012.

[35] D. Fahland and W. M. van der Aalst, "Model repair—aligning process models to reality," *Information Systems*, vol. 47, pp. 220–243, 2015.

[36] A. Burattin, A. Sperduti, and M. Veluscek, "Business models enhancement through discovery of roles." in *CIDM*, 2013, pp. 103–110.

[37] P. Jaisook and W. Premchaiswadi, "Time performance analysis of medical treatment processes by using disco," in *13th Int. Conference on ICT and Knowledge Engineering*. IEEE, 2015, pp. 110–115.

APPENDIX

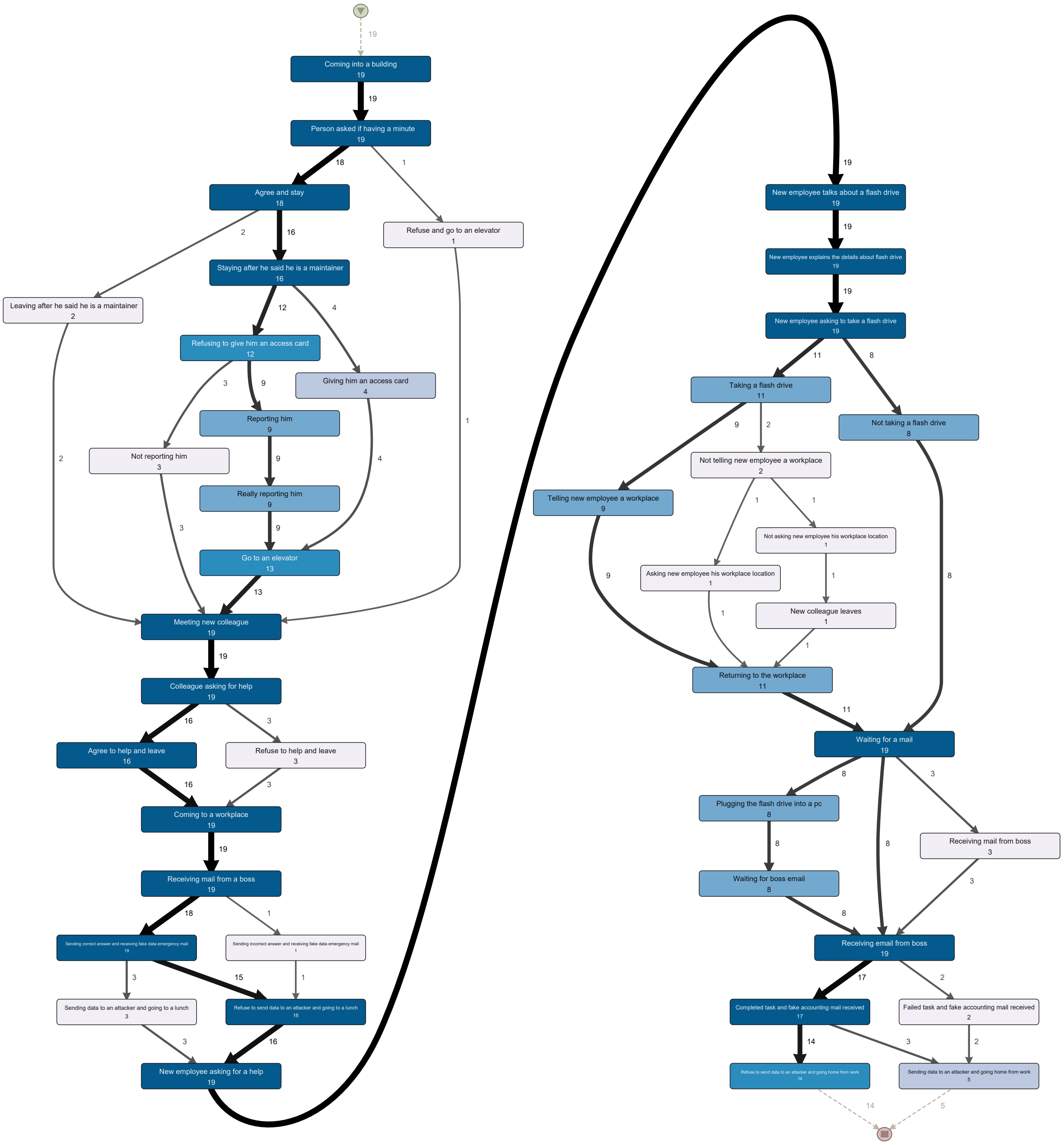

19
Coming into a building
19
19
Person asked if having a minute
19
18
1
Agree and stay
18
2
16
Refuse and go to an elevator
1
Staying after he said he is a maintainer
16
Leaving after he said he is a maintainer
2
12
4
Refusing to give him an access card
12
3
9
Giving him an access card
4
Reporting him
9
1
Not reporting him
3
2
9
4
Really reporting him
9
3
9
Go to an elevator
13
13
Meeting new colleague
19
19
Colleague asking for help
19
16
3
Agree to help and leave
16
Refuse to help and leave
3
16
3
Coming to a workplace
19
19
Receiving mail from a boss
19
18
1
Sending correct answer and receiving fake data emergency mail
19
Sending incorrect answer and receiving fake data emergency mail
1
3
15
1
Sending data to an attacker and going to a lunch
3
Refuse to send data to an attacker and going to a lunch
16
3
16
New employee asking for a help
19
19
New employee talks about a flash drive
19
19
New employee explains the details about flash drive
19
19
New employee asking to take a flash drive
19
11
8
Taking a flash drive
11
Not taking a flash drive
8
9
2
Not telling new employee a workplace
2
Telling new employee a workplace
9
1
1
Not asking new employee his workplace location
1
9
Asking new employee his workplace location
1
1
8
New colleague leaves
1
1
1
Returning to the workplace
11
11
Waiting for a mail
19
8
3
Plugging the flash drive into a pc
8
Receiving mail from boss
3
8
8
Waiting for boss email
8
3
8
Receiving email from boss
19
17
2
Completed task and fake accounting mail received
17
Failed task and fake accounting mail received
2
14
3
2
Refuse to send data to an attacker and going home from work
14
Sending data to an attacker and going home from work
5
14
5

Fig. 11: Discovered process model of the played game

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 357–361
DOI: 10.15439/2021F114
ISSN 2300-5963 ACSIS, Vol. 25

Matrix profile for DDoS attacks detection

Faisal Alotaibi
Department of Computer Science
University of Liverpool

Email:Faisal.alotaibi@liverpool.ac.uk

Alexei Lisitsa
Department of Computer Science
University of Liverpool

Email: A.Lisitsa@liverpool.ac.uk

***Abstract*—Several previous studies have focused on Distributed Denial of Service (DDoS) attacks, which are a crucial problem in computer network security. In this paper we explore the applicability of a a time series method known as a matrix profile to the anomaly based DDoS attacks detection. The study thus examined how the matrix profile method performed in diverse situations related to DDoS attacks, as well as identifying those features that are most applicable in various scenarios. Based on reported empirical evaluation the matrix profile method is shown to be efficient against most of the considered types of DDoS attacks.**

I. Introduction

THE Internet has grown at an exponential rate since the 1960s [1], and 3 billion people now surf the Internet every day to access social media, banking, shopping, and other everyday services [1]. However, the Internet is not a safe zone, and privacy and information security are major causes for concern. Any system connected to the Internet is subject to security threats from hackers, viruses, or sniffers [2]. The most common approach to degrading the availability of a targeted service on the Internet is a Distributed Denial of Service (DDoS) attack. DDoS attacks can range from the misuse of application-level vulnerabilities to high-volume flooding on a network [3] , and they are undoubtedly one of the leading causes of concern for many companies, organisations, and institutions [1]. A DDoS attack may thus refer to any malicious coordinated attack against any form of online services, whether these are commercial websites, bank websites, or government websites. A DDoS attack is usually performed by a massive number of bots over a specified period, either flooding a network with high volumes of irrelevant data to create excess traffic or attacking a vulnerable application to render it useless [4]. Although it is often easy to probe service availability and decongest the network, the most significant challenge in assessing such attacks lies in differentiating between legitimate congestion and attacker-initiated congestion, however, as these may manifest in similar ways [5].

There are many types of DDoS attacks, though these can be summarised as follows:

- Value Based Attacks. Such attacks include i) User Datagram Protocol (UDP) floods, ii) Ping Floods, and iii) Spoofed-packet floods.
- Protocol Based Attacks. Such attacks include i) SYN Floods, ii) fragmented packet attacks, iii) “Ping of Death”, and iv) Smurf DDoS.
- Application Layer Attacks. Such attacks include i) low-and-slow attacks, ii) GET/POST floods, iii) attacks that target Apache, iv) attacks that target Windows, and vi) OpenBSD vulnerabilities.

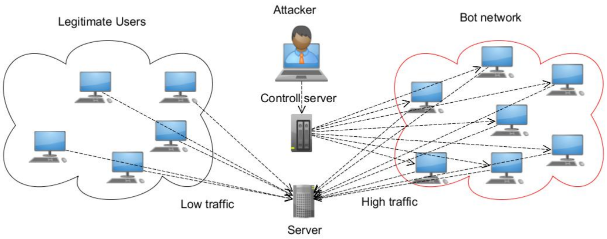


Fig. 1. DDoS attacks

As Cisco reports,The DDoS is predicted to become more harmful in the future and the world needs to develop appropriate solutions for the many scenarios that could arise. Over the next few years, all forms of DDoS attacks are likely to become more common, with predictions suggesting that the total number of DDoS attacks will double from the 7.9 million seen in 2018 to over 15 million by 2023.

Research on DDoS attacks detection and mitigation has proposed many efficient solutions [6]–[8]. Still due to unprecedented scale of the threat, a need for new highly scalable and precise solutions remains high. In this paper we present our initial study on the applicability of very powerful and promising approach in time series data mining, *matrix profile* [9], method for the detection of DDoS attacks. The paper is organized as follows. In the next section we present the basics of the matrix profile (MP) method, anomaly based detection using MP, dataset used in the experiments and data pre-processing. In the following section we discuss the details of the implementation. Section IV presents the results and discussion. Section V concludes the paper.

II. Matrix profile

The Matrix profile is a method, including a data structure and very efficient algorithms for computing *all-pairs-similarity-search* (or similarity join) for time series subsequences [10]. Since its invention, matrix profile has been shown to be a powerful method for solving various tasks in

IEEE Catalog Number: CFP2185N-ART ©2021, PTI

time series data mining including motif discovery, classification and anomaly detection among others [9], [11], [12]. The idea of matrix profile is very natural. For a time series $T = t_0, \dots t_n$ and a positive integer m denoted by $T_{i,m}$ a subsequence $t_i, \dots, i_{i+m-1}$ of T. The matrix profile of T includes the following data: 1) distance profile which is a vector of distances between all pairs of subsequences $T_{i,m}$ and $T_{j,m}$ of T; 2) profile index which for every i stores j such that $T_{j,m}$ is the closest to $T_{i,m}$ among all m-subsequences ("a distance to the nearest neighbour"). While any concept of distance/metrics can be used in matrix profile, the standard euclidean distance between z-normalized values is a common choice [9], [11], [12]. The advantages of the matrix profile method include its support for very efficient and highly parallelizable algorithms for similarity join, the fact that it is domain agnostic, that fact that it offers precise solutions and requires only a single parameter (but can be expanded to multi-dimensional cases as well). Yet another crucial feature for many applications of matrix profile is that it supports *incremental* algorithms, so it can be applied for online processing.

Algorithm 1 Matrix profile

```
procedure MATRIX PROFILE($T$, $m$)
    $n \leftarrow$ length of ($T$),   $l \leftarrow n\text{-}m\text{+}1$
    $\mu, \sigma \leftarrow ComputeMeanStd(T,m)$
    $QT \leftarrow SlidingDotProduct(T[1:m], T)$
    $QT_{first} \leftarrow QT$
    $D \leftarrow CalculateDistanceProfile(QT, \mu, \sigma)$
    $P \leftarrow D$,   $I \leftarrow ones$
    for $i = 2$ to $l$ do
        for $j = l$ downto $2$ do
            $QT[j] \leftarrow QT[j-1] - T[j-1] \cdot T[i-1] + T[j+m-l] \cdot T[j+m-l]$
        end for
        $QT[1] \leftarrow QT_{frist}[i]$
        $D \leftarrow CalculateDistanceProfile(QT,\ \mu,\ \sigma,\ i)$
        $P,\ I \leftarrow ElementWiseMin(P,\ I,\ D,\ i)$
    end for
    Return $P, I$
end procedure
```

Time series discords, that is subsequences with the large (maximal) distances to their nearest neighbours have already been proposed as novelty/anomaly detectors [9] and they can be easily identified using matrix profile data. Indeed, one has just to check the values of $\mid p(i) - i \mid$, where $p(i)$ is a profile index value for i. Thus, if we consider the metric used in matrix profile as a *similarity measure*, an *anomalous subsequence* is the one for which the most similar subsequence is found far away. Such an approach for anomaly detection has been considered already in [13], [14] for the medical domain.

A. Anomaly based detection with MP

In this study we investigate the applicability of matrix profile method in computer networks security domain, in particular, for anomaly based intrusion detection.

The outline of the proposed generic scheme for such a detection is as follows. We fix a time window W, subsequences length M and threshold value T.

- The traffic data is converted into time series;
- Matrix Profile method is applied to the subsequences of length M of the time series;
- If MP value for a given time window is greater than T then an anomaly is detected, if it is lower than T, no anomaly is detected and the traffic is considered as normal.

The implementation of such a scheme requires some choices to be made. The conversion of traffic data into time series can be done in various ways using different features of the data. Depending on the format of the data further processing might be needed as well. Finally the choices of the time window and threshold value have to be made.

We have focused on detection of DDoS attacks and have used for experiments a DDoS Evaluation Dataset (CIC-DDOS2019) obtained from the Canadian Institute for Cybersecurity[1] . This dataset is fully labelled, which helps in terms of measuring the performance matrix profile based on making comparisons with the times when attacks take place. We have conducted the experiments in offline scenario, while the case of online processing is a topic of our ongoing research.

B. Dataset CIC-DDoS2019

This dataset can be publicly accessed and includes two data formats: the first is pcap, while the second is CSV. While the pcap files include raw data recorded over two days, the CSV files include the information on network flows extracted using CICFlowMeter-V3. There are 80 variable features available. The features used in this work included Total Fwd Packets, Total Bwd Packets, Total Length of Fwd Packets, Total Length of Bwd Packets, Fwd Packet Length, Max, Fwd Packet Length, Min, 'Subflow Fwd Packets', Fwd Packet Length Mean, Fwd Packet Length Std, 'Bwd Avg Bulk Rate, and Bwd Packet Length Max.

Each CSV file contains a label for the flow that describe the flow type (normal or named for the nature of the attack); thus, for each type of attack, there is a separate CSV file.

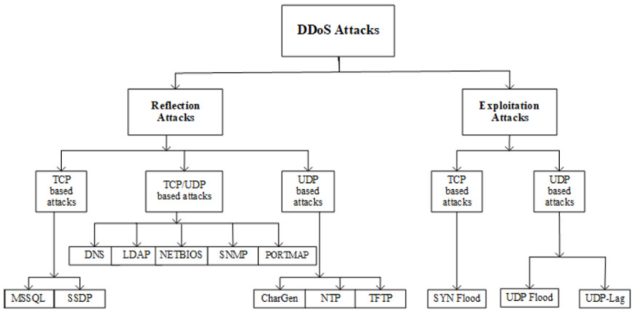
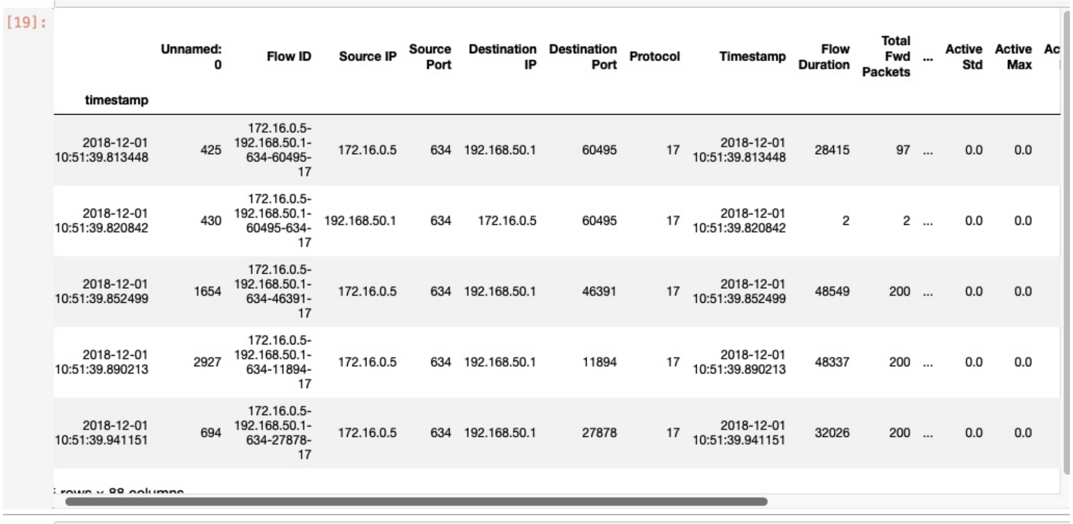


Fig. 2. Dataset for DDoS (https://www.unb.ca/cic/datasets/ddos-2019.html)

C. Data pre-processing

Fig. 3 shows the data sample used in this implementation. The data set file includes a time stamp indicating the time of

[1] https://www.unb.ca/cic/datasets/ddos-2019.html

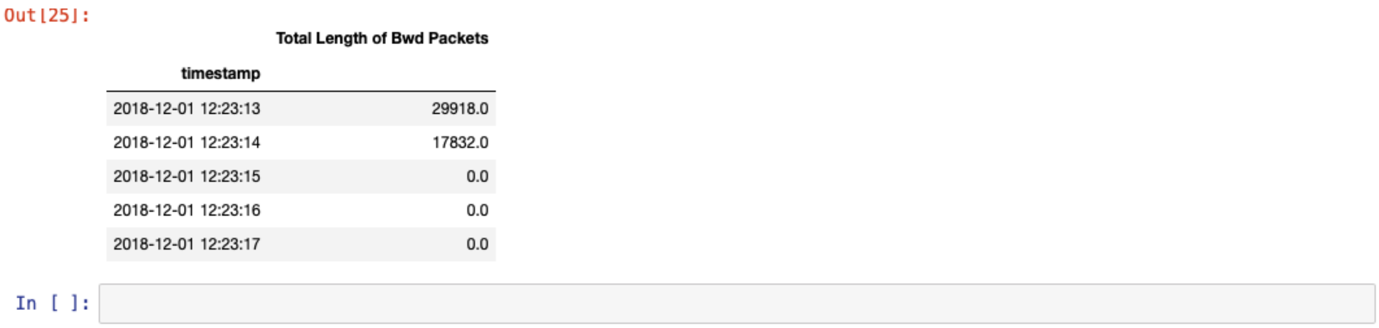


[19]:

	Unnamed: 0	Flow ID	Source IP	Source Port	Destination IP	Destination Port	Protocol	Timestamp	Flow Duration	Total Fwd Packets	...	Active Std	Active Max	Ac
timestamp														
2018-12-01 10:51:39.813448	425	172.16.0.5-192.168.50.1-634-60495-17	172.16.0.5	634	192.168.50.1	60495	17	2018-12-01 10:51:39.813448	28415	97	...	0.0	0.0	
2018-12-01 10:51:39.820842	430	172.16.0.5-192.168.50.1-60495-634-17	192.168.50.1	634	172.16.0.5	60495	17	2018-12-01 10:51:39.820842	2	2	...	0.0	0.0	
2018-12-01 10:51:39.852499	1654	172.16.0.5-192.168.50.1-634-46391-17	172.16.0.5	634	192.168.50.1	46391	17	2018-12-01 10:51:39.852499	48549	200	...	0.0	0.0	
2018-12-01 10:51:39.890213	2927	172.16.0.5-192.168.50.1-634-11894-17	172.16.0.5	634	192.168.50.1	11894	17	2018-12-01 10:51:39.890213	48337	200	...	0.0	0.0	
2018-12-01 10:51:39.941151	694	172.16.0.5-192.168.50.1-634-27878-17	172.16.0.5	634	192.168.50.1	27878	17	2018-12-01 10:51:39.941151	32026	200	...	0.0	0.0	

rows × 88 columns

Fig. 3. Dataset Before conversion



Out[25]:

	Total Length of Bwd Packets
timestamp	
2018-12-01 12:23:13	29918.0
2018-12-01 12:23:14	17832.0
2018-12-01 12:23:15	0.0
2018-12-01 12:23:16	0.0
2018-12-01 12:23:17	0.0

In [ ]:

Fig. 4. Dataset after conversion

the start of the flow. In order to use this data set in a matrix profile it has to be converted. The first step is to aggregate the data set for the traffic based on the time window: different flows have different times, and it is important to sum or group all the flows for each feature.

In this work, we have applied the following form of aggregation. For a chosen feature and for each time window we consider *all flows which start in that window* and aggregate the feature values across all these flows. For numerical features the summation is used as an aggregation operation.

As an example, Figure 4 shows the result of converting the data where `Total Length of Bwd Packets` is used as a feature. As most of the data in the data set were attacks, an additional normal traffic was also added to the data set; this was added thirty minutes before the attack start and after the attack end.

III. Implementation

Our implementation has proceeded by following steps.

1) Reading the dataset
2) Increasing normal traffic
3) Feature selection
4) Resampling traffic (time window aggregation)
5) Running data in matrix profile mode
6) Processing the output for each threshold
7) Measuring performance .
8) Repeating the experiment with different features and different attacks

This section offers details for each step in the implementation process. In this experiment, several attacks were assessed on both day 1 and day 2, as shown in table 2. The CSV file consists of different traffic flows including labels, with two types of labels (normal or attack). Most of the flows in the csv file as downloaded were attacks. Normal traffic in this dataset is labelled (BEIGN), while attacks are each named after the specific type of attack. As attacks dominated the traffic in the dataset, an increase in normal traffic was required before beginning the experiment for the following reasons:

1) The matrix profile works to identify anomalies, which must thus anomaly be unusual events; a dataset where attacks seem to be the norm is thus inappropriate for attack detection.
2) In real network scenarios, the normal traffic should dominate the anomalous traffic rather than the other way around.

TABLE I
RESULTS FOR EXPERIMENT THRESHOLD 0.5 FOR DAY 1

Distributed denial of service attacks				
Day	Attack	Threshold	Features	Accuracy
1	DrDNS	0.5	All	66%
1	LDAP	0.5	Fwd Packet Length Std	86%
1	MSSQL	0.5	Fwd Packet Length Std	80%
1	NETBIOS	0.5	Fwd Packet Length Std	82%
1	NTP	0.5	All	38%
1	SNMP	0.5	All	72%
1	SSDP	0.5	Total Length of Bwd Packets	82%
1	UDP	0.5	Fwd IAT MEAN	69%
1	SYN	0.5	Fwd packets Length Std	93%
1	TFTP	0.5	All	93%
1	UDPLag	0.5	All	69%

TABLE II
RESULTS FOR EXPERIMENT THRESHOLD 1 FOR DAY 1

Distributed Denial of service attacks				
Day	Attack name	threshold value	features	Accuracy
1	DrDNS	1	All features	66%
1	LDAP	1	Fwd IAT min	88%
1	MSSQL	1	Syn flag count	86%
1	NETBIOS	1	Bwd IAT Std	82%
1	NTP	1	All features	38%
1	SNMP	1	All features	72%
1	SSDP	1	Bwd IAT Max	81%
1	UDP	1	Fwd IAT Std	68%
1	SYN	1	All features	93%
1	TFTP	1	All features	82%
1	UDPLag	1	All features	69%

TABLE III
RESULTS FOR EXPERIMENT THRESHOLD 2 FOR DAY 1

Distributed Denial of service attacks				
Day	Attack name	threshold value	features	Accuracy
1	DNS	2	All features	66%
1	LDAP	2	All features	86%
1	MSSQL	2	All features	80%
1	NETBIOS	2	All features	82%
1	NTP	2	All features	38%
1	SNMP	2	All features	72%
1	SSDP	2	All features	81%
1	UDP	2	All features	68%
1	SYN	2	All features	93%
1	TFTP	2	All features	82%
1	UDPLag	2	All features	69%

Normal traffic was thus increased to make it the most common. As the attack duration in each case was around 10 to 15 minutes, similar steps were followed in each case: to increase the normal traffic, all the normal traffic available in a given dataset was duplicated multiple times; the resulting new normal traffic block was then inserted 30 minutes before the attack began, with a random time function to change the distribution within that 30 minutes. This was repeated for the 30 minutes after the attack ended in each case. This created datasets dominated by normal traffic.

The next step was to select features one by one, as the matrix profile accepts only one dimension. It was thus necessary to run the experiment for each feature separately. Re-sampling of the traffic to deliver time window aggregation was required after feature selection, based on the time window required.

The time window used in the experiments was one second. Further choices were 1) the length of the subsequences used in Matrix Profile set to M=10; 2) threshold MP values tested were 0.5, 1, 2.

The performance of the detection procedure was measured in terms of detection precision using labelled data in the dataset as the source of ground truth. The detection event is considered as *true positive* if the anomaly in a time window was detected and there was at least one flow starting in that window which is labelled as an attack.

In this study, a Python 3 library from the Matrix Profile Foundation was used to perform Matrxi Profile computations.[2]

IV. Results and Discussion

We conducted the experiments and created confusion matrices for each combination of an attack, chosen features and chosen threshold values. The results were then assessed against the following criteria.

1) Success: Where the confusion matrix accuracy for each threshold value in each feature is over 70%, it is considered to represent a successful detection. This occurred for LDAP, MSSQL, NETBIOS, SSDP, SYN and TFTP in the day one results.
2) Struggling: Where the confusion matrix accuracy for each threshold value in each feature is 50% to 70% inclusive, the detection cannot be considered good. This occurred for the portmap attack, where detection showed 57% accuracy.
3) Failure: When the confusion matrix accuracy for each threshold value in each feature is 50% or lower, this is considered as a failure of detection, as seen in the day one attack NTP, which had an accuracy of only 38%.

After all the experiments were completed, the best result for each attack at each threshold value was recorded. As seen in the tables:

- The average of all accuracy result in threshold value 0.5 is 64.68%
- The average of all accuracy result in threshold value 1.0 is 67.84%
- The average of all accuracy result in threshold value 2.0 is 74.53%

Different threshold values produce different accuracy results. Consequently, based on our experiment we suggest that the optimal threshold value to be 2.0.

Finally, we notice, based on the literature review that thus far no study have used any unsupervised learning method with this dataset. However, there have been some works that used supervised learning exemplified in [15]. Their method successfully achieved an accuracy of 99%. While this result is typical in supervised learning, our work is different. First of all, we use unsupervised processing/learning based on matrix profile. Second, Matrix profile only accepts one-dimensional data . Further to that our pre-processing of the data is done to increase normal traffic in the dataset which simulates how

[2] https://pypi.org/project/matrixprofile/

TABLE VI
RESULTS FOR EXPERIMENT THRESHOLD 2 DAY 2

Distributed Denial of service attacks				
attack day	Attack name	threshold value	features	Accuracy
2	LDAP	0.5	All features	74%
2	MSSQL	0.5	All features	82%
2	NETBIOS	0.5	All features	89%
2	portmap	0.5	All features	57%

TABLE IV
RESULTS FOR EXPERIMENT THRESHOLD 0.5 FOR DAY 2

Distributed Denial of service attacks				
attack day	Attack name	threshold value	features	Accuracy
2	LDAP	0.5	All features	74%
2	MSSQL	0.5	All features	82%
2	NETBIOS	0.5	All features	89%
2	portmap	0.5	All features	57%

TABLE V
RESULTS FOR EXPERIMENT THRESHOLD 1 DAY 2

Distributed Denial of service attacks				
attack day	Attack name	threshold value	features	Accuracy
2	LDAP	0.5	All features	74%
2	MSSQL	0.5	All features	82%
2	NETBIOS	0.5	All features	89%
2	portmap	0.5	All features	57%

DDoS attacks normally happen in the network. Also we give more details of the detection of different types of DDoS attack while other studies treated all DDoS attack as one group.

V. CONCLUSION

This study aimed to examine the advantages of using a Matrix Profile algorithm to address network security problems with DDoS attacks. The results of this initial study showed that this method is effective against multiple specific types of DDoS attacks. The next step will be to develop a module to allow the Matrix Profile method to be used online and the resulting performance to be assessed. Broader classes of settings should be explored and the detection of wider classes of attacks should be considered.

REFERENCES

[1] Dhruba Kumar Bhattacharyya and Jugal Kumar Kalita. Ddos attacks evolution, detection, prevention, reaction, and tolerance. 2016.

[2] Zhuo Lin. Internet security and firewall [j]. *Journal of Changsha University*, 15:32–35, 2001.

[3] Susan J Harrington. Why people copy software and create computer viruses. *Information Resources Management Journal (IRMJ)*, 2(3):28–38, 1989.

[4] Neelam Dayal and Shashank Srivastava. Analyzing behavior of DDoS attacks to identify DDoS detection features in SDN. In *2017 9th International Conference on Communication Systems and Networks (COMSNETS)*, pages 274–281. IEEE, jan 2017.

[5] Shibo Luo, Jun Wu, Jianhua Li, and Bei Pei. A Defense Mechanism for Distributed Denial of Service Attack in Software-Defined Networks. *9th International Conference on Frontier of Computer Science and Technology (FCST 2015)*, pages 325–329, 2015.

[6] Chirag Modi, Dhiren Patel, Bhavesh Borisaniya, Hiren Patel, Avi Patel, and Muttukrishnan Rajarajan. A survey of intrusion detection techniques in cloud. *Journal of network and computer applications*, 36(1):42–57, 2013.

[7] Kevin Hoffman, David Zage, and Cristina Nita-Rotaru. A survey of attack and defense techniques for reputation systems. *ACM Computing Surveys (CSUR)*, 42(1):1–31, 2009.

[8] Dhruba Kumar Bhattacharyya. *DDoS attacks: evolution, detection, prevention, reaction, and tolerance*. Chapman and Hall/CRC, 2019.

[9] Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. Matrix profile i: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In *2016 IEEE 16th international conference on data mining (ICDM)*, pages 1317–1322. Ieee, 2016.

[10] Shaghayegh Gharghabi, Shima Imani, Anthony Bagnall, Amirali Darvishzadeh, and Eamonn Keogh. Matrix profile xii: Mpdist: a novel time series distance measure to allow data mining in more challenging scenarios. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 965–970. IEEE, 2018.

[11] Dieter De Paepe, Sander Vanden Hautte, Bram Steenwinckel, Filip De Turck, Femke Ongenae, Olivier Janssens, and Sofie Van Hoecke. A generalized matrix profile framework with support for contextual series analysis. *Eng. Appl. Artif. Intell.*, 90(C), April 2020.

[12] Frank Madrid, Shima Imani, Ryan Mercer, Zachary Zimmerman, Nader Shakibay, and Eamonn Keogh. Matrix profile xx: Finding and visualizing time series motifs of all lengths using the matrix profile. In *2019 IEEE International Conference on Big Knowledge (ICBK)*, pages 175–182. IEEE, 2019.

[13] Haemwaan Sivaraks and Chotirat Ratanamahatana. Robust and accurate anomaly detection in ecg artifacts using time series motif discovery. *Computational and mathematical methods in medicine*, 2015:453214, 01 2015.

[14] Rutuja Wankhedkar and Sanjay Kumar Jain. Motif discovery and anomaly detection in an ecg using matrix profile. In *Progress in Advanced Computing and Intelligent Engineering*, pages 88–95. Springer, 2021.

[15] Mahmoud Said Elsayed, Nhien-An Le-Khac, Soumyabrata Dev, and Anca Delia Jurcut. Ddosnet: A deep-learning model for detecting network attacks. In *2020 IEEE 21st International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*, pages 391–396. IEEE, 2020.

Advances in Information Systems and Technology

AIST is a FedCSIS conference track aiming at integrating and creating synergy between disciplines of information technology, information systems, and social sciences. The track addresses the issues relevant to information technology and necessary for practical, everyday needs of business, other organizations and society at large. This track takes a socio-technical view on information systems and, at the same time, relates to ethical, social and political issues raised by information systems.

AIST provides a forum for academics and professionals to share the latest developments and advances in the knowledge and practice of these fields. It seeks new studies in many disciplines to foster a growing body of conceptual, theoretical, experimental, and applied research that could inform design, deployment and usage choices for information systems and technology within business and public organizations as well as households.

We call for papers covering a broad spectrum of topics which bring together sciences of information systems, information technologies, and social sciences, i.e., economics, management, business, finance, and education. The track bridges the diversity of approaches that contributors bring to the conference. The main topics covered are:

- Advances in information systems and technologies for business;
- Advances in information systems and technologies for governments;
- Advances in information systems and technologies for education;
- Advances in information systems and technologies for healthcare;
- Advances in information systems and technologies for smart cities; and
- Advances in information systems and technologies for sustainable development.

AIST invites papers covering the most recent innovations, current trends, professional experiences and new challenges in the several perspectives of information systems and technologies, i.e. design, implementation, stabilization, continuous improvement, and transformation. It seeks new works from researchers and practitioners in business intelligence, big data, data mining, machine learning, cloud computing, mobile applications, social networks, internet of thing, sustainable technologies and systems, blockchain, etc.

Extended versions of high-marked papers presented at technical sessions of AIST 2015-2020 have been published with Springer in volumes of Lecture Notes in Business Information Processing: LNBIP 243, LNBIP 277, LNBIP 311, LNBIP 346, and LNBIP 380.

Extended versions of selected papers presented during AIST 2021 will be published in Lecture Notes in Business Information Processing series(LNBIP, Springer).

- Data Science in Health, Ecology and Commerce (3rd Special Session DSH'21)
- Information Systems Management (16th Conference ISM'21)
- Knowledge Acquisition and Management (27th Conference KAM'21)

Track Chairs

- **Ziemba, Ewa,** University of Economics in Katowice, Poland
- **Chmielarz, Witold,** University of Warsaw, Poland
- **Cano, Alberto,** Virginia Commonwealth University, Richmond, United States

Program Chairs

- **Chmielarz, Witold,** University of Warsaw, Poland
- **Raban, Daphne,** University of Haifa, Israel
- **Wątróbski, Jarosław,** University of Szczecin, Poland
- **Ziemba, Ewa,** University of Economics in Katowice, Poland

Program Committee

- **Anton Agafonov,** Samara National Research University, Russia
- **Andrzej Białas,** Institute of Innovative Technologies EMAG, Poland
- **Ofir Ben-Assuli,** Ono Academic College, Israel
- **Robertas Damasevicius,** Silesian University of Technology, Poland
- **Gonçalo Dias,** University of Aveiro, Portugal
- **Rafal Drezewski,** AGH University of Science and Technology, Poland
- **Leila Halawi,** Embry-Riddle Aeronautical University, United States
- **Ralf Haerting,** Hochschule Aalen, Germany
- **Adrian Kapczyński,** Silesian University of Technology, Poland
- **Wojciech Kempa,** Silesian University of Technology, Poland
- **Agnieszka Konys,** West Pomeranian University of Technology, Szczecin, Poland
- **Eugenia Kovatcheva,** University of Library Studies and Information Technologies, Bulgaria
- **Jan Kozak,** University of Economics in Katowice, Poland
- **Marcin Lawnik,** Silesian University of Technology, Faculty of Applied Mathematics, Poland

DOI: 10.15439/2021F109
ISSN 2300-5963 ACSIS, Vol. 25

Traffic Signal Control: a Double Q-learning Approach

Anton Agafonov
Samara National Research University
Samara, Russia
Email: ant.agafonov@gmail.com

Vladislav Myasnikov
Samara National Research University
Samara, Russia
Email: vmyas@geosamara.ru

***Abstract*—Currently, the use of information and communication technologies for solving economic, social, transportation, and other problems in the urban environment is usually considered within the "smart city" concept. Optimal traffic management and, in particular, traffic signal control is one of the key components of smart cities. In this paper, we investigate the reinforcement learning approach, namely, the double Q-learning approach, to solve the traffic signal control problem. Both the initial data on the connected vehicles distribution and the aggregated characteristics of traffic flows are used to describe the state of the reinforcement learning agent. Experimental studies of the proposed model were carried out on synthetic and real data using the CityFlow microscopic traffic simulator.**

I. Introduction

THE growth of the urbanization level poses the problems of increasing the efficiency of the urban resources and existing infrastructure usage. The amount of collected urban environment data and the development of information and communication technologies (ICT) are key factors in solving these problems [1]. The concept of city transformation using ICT is commonly referred to as a "smart city". Smart cities involve the use of a wide stack of information and communication technologies to solve economic, social, transportation, and other problems. A wide area of research, as a result, attracts the attention of scientists from different scientific fields who consider certain aspects of smart cities: smart mobility, smart urban environment, smart government, etc. [2].

Smart cities provide new opportunities to solve urban traffic management problems, optimize traffic flows and individual vehicle routes, reduce traffic congestion, environmental emissions, improve road safety, etc. [3], [4], [5], [6]. The development of connected devices and the Internet of Things, in general, is an important factor to make smart cities efficient in various aspects [7]. Moreover, one of the dominant trends in the development of modern intelligent transportation systems is the development of communication networks (VANET), and, as a consequence, the development of connected vehicles. Connected vehicles are vehicles that can communicate with other vehicles (V2V communications), infrastructure (V2I), and other road users (V2X). The exchange of information between road infrastructure and vehicles in real time can be used to improve the efficiency of traffic management, including through coordinated optimization of traffic signals and vehicle trajectories [8], [9].

In this paper, we consider the traffic signal control problem using information from connected vehicles in order to minimize the total travel time in the transport network. To solve this problem, it is proposed to use a reinforcement learning approach, in particular, a double Q-learning algorithm.

The work is structured as follows. Section II provides a literature review and describes classic and state-of-the-art traffic signal control methods. Section III introduces the basic notation and problem statement. In Section IV, we present a traffic signal control method based on a reinforcement learning approach. Experimental studies of the proposed method are described in Section V. Finally, we give some conclusions and possible directions for further research.

II. Related Work

In [10], the authors presented an overview of widely acknowledged classical transportation approaches and the current state of research on the traffic signal control problem. In [11], the authors analyzed the literature for 2015-2020 on the topic of traffic management, reviewed approaches based on microsimulation and computational intelligence, presented research gaps and possible directions for future work. An overview of traffic control methods using data from autonomous and connected vehicles is presented in [12]. The authors explained the advantages and disadvantages of different types of traffic control methods and discussed possible future research directions.

An overview of classic traffic signal control strategies is presented in [13]. For each traffic light, the control plan usually includes stage (or phase), phase split, cycle time, and offset. Fixed-time strategies use a control plan based on historical traffic data [14], [15]. State-based strategies determine the optimal cycle time and phase split, minimizing the total delay or maximizing the capacity of the intersection. Phase-based strategies further optimize the optimal staging for the intersection.

Separately, we can distinguish a class of strategies that apply coordinated traffic signals control at intersections in a certain area or the whole network. The MAXBAND algorithm [16] optimizes the phase displacement of traffic lights at adjacent intersections to maximize the number of vehicles that can

The work was supported by the Russian Science Foundation grant No. 21-11-00321, https://rscf.ru/en/project/21-11-00321/

pass through intersections without stopping. The TRANSYT method [17] uses a dynamic network model to iteratively select values of decision parameters, evaluate performance, and select the best set of parameters. In [18], the authors proposed an approach aimed at stabilizing demand and reducing the risk of oversaturation by balancing the queue length at the intersection. Optimization methods for urban-traffic management was applied in [19].

Most modern scientific research is devoted to the use of machine learning and artificial intelligence methods for solving the traffic control problem, and, in particular, reinforcement learning approaches. In [20], the authors reviewed various reinforcement learning models and algorithms applied to traffic signal control, classified by model characteristics (state space, actions, rewards) and performance metrics. An analysis of modern deep reinforcement learning approaches for the adaptive traffic signal control problem is presented in [21]. The authors provided recommendations for adequate model choice, architecture design, and hyper-parameters tuning. In [22], the authors compared traffic optimization methods with different Q-learning approaches and different objective functions but considered only a single intersection environment.

In [23], the authors used a Q-learning approach, training a separate reinforcement learning agent for each intersection independently, without considering information at adjacent intersections. The authors' research was continued in [24], [25]. In [24], the authors used the state of the entire network to train the graph attention network that controls all intersections. However, using the data of the entire network in the feature vector significantly increases the training time and the amount of required memory. In [25] it was proposed to use the concept of "pressure" to achieve coordinated control in the network.

In [26], the authors investigated a multi-agent algorithm based on Q-learning, taking into account the traffic state at neighboring intersections. In [27], the authors proposed using a knowledge exchange protocol between agents to increase the level of cooperation between agents and achieve an optimal traffic light control strategy. A double Q-learning algorithm for improving the stability of control policy was investigated in [28]. In [29], the authors combined the recurrent neural network (RNN) with Deep Q-Network and showed that the proposed approach performs better in partially observed environment. However, the experimental study was conducted at only one intersection.

In this paper, we consider a double Q-learning model in which one agent is trained on the data from all considered intersections. As a vector for describing the network state, both the initial information about the distribution of vehicles by lanes and the aggregated characteristics of the traffic flow (queue length at the intersection, pressure) are used. The experimental study of the proposed solution was conducted both on synthetic and real-world datasets.

The next section provides basic notation and problem statement.

III. Problem Statement

In this paper, we consider the traffic signal control problem. Each intersection in the transportation network is controlled by a reinforcement learning agent that chooses an action based on the observed state on the intersection. To decrease the computational complexity, we train one Q-learning neural network. It means that all the agents share the same neural network.

The traffic signal control problem as a reinforcement learning problem is usually presented as a Markov decision process that can be defined by a tuple $\langle \mathbf{S}, \mathbf{A}, \mathbf{P}_a, \mathbf{R}_a \rangle$, where:

- $\mathbf{S}$ is the system state space,
- $\mathbf{A}$ is the action space,
- $\mathbf{P}_a(s, s') = Pr\left(s_{t+1} = s' | s_t = s, a_t = a\right)$ is the transition of probability from state s to state s' under the action a at time t,
- $\mathbf{R}_a\left(s, s'\right)$ is the immediate reward after the transition from state s to state s' under action a.

Let us consider these definitions in more detail in accordance with the considered traffic signal control problem.

It is assumed that each agent i at time step t observes a current system state $s^t \in \mathbf{S}$. In this paper, we consider the following factors that describe the environment:

- current traffic signal phase,
- queue length on each incoming lane,
- number of vehicles on each spatial segment of the incoming and outgoing lanes

Next, each agent chooses an action $a_t^i \in \mathbf{A}$ for the next time interval Δt. The chosen action set a^t of all agents is sent to the system that transit to a new state $s_{t+1} \in \mathbf{S}$ according to the transition probability. The reward $\mathbf{R}_{a_t}\left(s_t, s_{t+1}\right)$ is determined.

The main idea of the traffic signal control problem is to minimize the total travel time for all vehicles in the system. However, this is hard to optimize this criterion directly since the travel time metric cannot be used to calculate the instant reward after the transition in state s_{t+1}. In this paper, we calculate the reward for agent i as a weighted linear combination of several factors that indirectly describe the traffic situation:

$$r_t^i = \alpha_0 \sum_{l \in L^i} q_t^l + \alpha_1 \sum_{l \in L^i} v_t^l + \alpha_2 p^i, \tag{1}$$

where $\alpha_j, j = \overline{0,2}$ are the weight coefficients, L^i is the set of incoming lanes at the intersection i, q_t^l is the queue length on lane l at time t, v_t^l is the average speed of all vehicles on lane l at time t, p^i is the pressure [18], i.e. the difference between the incoming and outgoing number of vehicles at the intersection i.

The goal of the reinforcement learning problem is to learn a policy $\pi^i : \mathbf{A} \times \mathbf{S} \rightarrow [0,1]$, $\pi(a,s) = Pr(a_t = a | s_t = s)$ for each agent i that maximizes the expected cumulative reward:

$$R^i = \sum_{t=0}^{T} \gamma_t r_t^i, \tag{2}$$

where T is the total times steps number, $\gamma \in [0,1]$ is the discount factor.

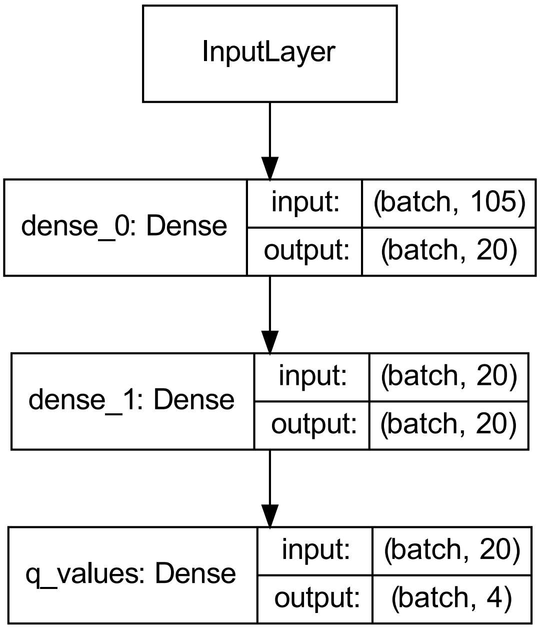


Fig. 1. Neural network architecture

IV. Methodology

To solve the traffic signal control problem as a reinforcement problem, we propose to use a double Q-learning approach that is used to overcome the problem of overestimating the action values in a noisy environment.

Consider the action-value function (Q-function) of a pair (s, a) under the policy π:

$$Q^{\pi}(s,a) = E\{R|s,a,\pi\}. \tag{3}$$

One of the possible solution to find the optimal policy π^* is to find the optimal Q-function:

$$Q^{*}(s,a) = \max_{\pi} Q^{\pi}(s,a). \tag{4}$$

In Q-learning, an iterative procedure is used:

$$\begin{aligned} Q^{new}(s_t,a_t) \leftarrow &(1-\alpha)Q(s_t,a_t)+ \\ &+\alpha\left(r_t+\gamma\max_a Q\left(s_{t+1},a\right)\right), \end{aligned} \tag{5}$$

where $\alpha \in (0,1]$ is a learning rate.

In the double Q-learning approach [30], two Q-functions Q^A, Q^B are used as a double estimator in the following way:

$$\begin{aligned} &Q^A_{t+1}(s_t,a_t) = (1-\alpha)Q^A_t(s_t,a_t)+ \\ &\quad +\alpha\left(r_t+\gamma Q^B_t\left(s_{t+1},\arg\max_a Q^A_t\left(s_{t+1},a\right)\right)\right), \\ &Q^B_{t+1}(s_t,a_t) = (1-\alpha)Q^B_t(s_t,a_t)+ \\ &\quad +\alpha\left(r_t+\gamma Q^A_t\left(s_{t+1},\arg\max_a Q^B_t\left(s_{t+1},a\right)\right)\right), \end{aligned} \tag{6}$$

In this paper, to approximate the Q-functions we use two neural networks with the same simple architecture that is shown in Fig. 1.

We train networks on the data from all intersections, so all the agents use the networks with the same parameters. The output value of the network is the action vector for one intersection.

In the next section, we present an experimental study of the proposed approach.

V. Experimental Study

To conduct an experimental study, we use an open-source traffic simulator CityFlow [31] designed for large-scale traffic scenarios. The simulator provides a Python interface to implement different modules. In particular, the simulator provides data access methods for obtaining information about the position/speed of each vehicle in the transport network, as well as control methods for setting the traffic signal phase, vehicle routes, etc.

We conduct our experiments on two datasets [24]:

- Synthetic 6×6 grid network dataset.
- Real-world data New York dataset that contains 196 intersections with traffic flow information from open-source taxi trip data.

We compare the proposed double Q-learning approach with the following classical and reinforcement learning methods:

- FixedTime [13] method that uses a predefined traffic signal phase plan with random offsets.
- MaxPressure [18] method that chooses that phase that maximizes the pressure at an intersection.
- Individual RL [23] method in which each intersection is controlled by an individual agent, each agent train and use a separate neural network.
- CoLight [24] method in which one agent is trained on data from the whole network and returns an action for each intersection.
- Double QL: considered in this paper double Q-learning algorithm.

Experiments were performed iteratively, in several runs. Each run consists of the following steps:

1) Perform a traffic simulation using trained (or default) Q-functions and store system states and reward values.
2) Create a training dataset using obtained system states/rewards.
3) Train Q-functions.
4) Calculate the average travel time in the network using the trained Q-functions.

To compare the effectiveness of the considered methods, we evaluate the average travel time of all vehicles in the network. The metric shows the average time that all vehicles spend to complete their trips from the origin to the destination. The performance comparison of the algorithms by the described criteria is presented in Table I.

The Individual RL method is not performed on the New York dataset due to memory limits.

The proposed double Q-learning approach showed the best results in comparison with baseline algorithms.

TABLE I
PERFORMANCE COMPARISON OF THE ALGORITHMS BY AVERAGE TRAVEL TIME

Model	Grid 6×6	NewYork
FixedTime	210.94	1826.78
MaxPressure	195.49	1225.97
Individual RL	171.97	-
CoLight	177.45	1316.04
Double QL	**165.71**	**1099.19**

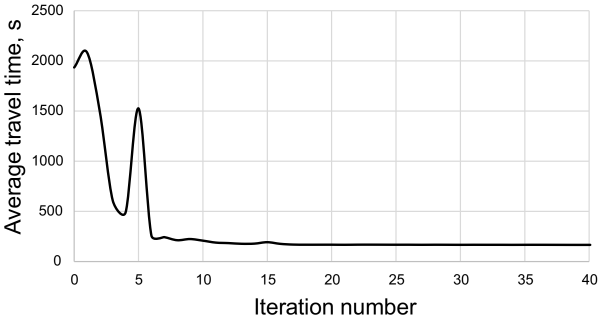


Fig. 2. Convergence speed on the 6×6 dataset

Finally, we estimate the convergence of the double Q-learning model. Fig. 2 shows the convergence speed on the synthetic dataset, Fig. 3 - on the New York dataset.

The model starts with the high average travel time value that decreases during iterations. For the synthetic network, the average travel time reaches a stable optimal value very fast; for the New York dataset, the convergence is worse.

VI. CONCLUSION

In this paper, we consider the double Q-learning algorithm to solve the traffic signal control problem. It is supposed, that the problem is solved in the connected environment, where position/speed information is available for each vehicle. This information was used to describe the system state in the reinforcement learning problem statement. The proposed approach was evaluated using the microscopic traffic simulation. Experimental analysis on synthetic and real-world traffic data allows us to conclude that the considered method outperforms other classical and reinforcement learning algorithms.

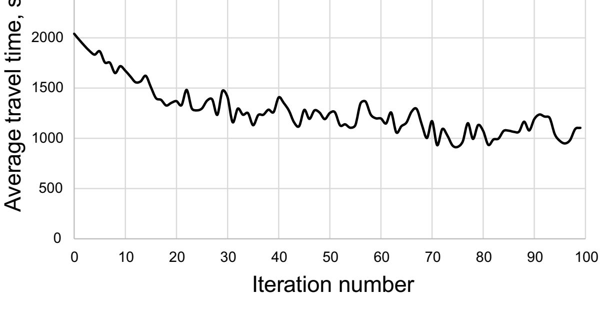


Fig. 3. Convergence speed on the New York dataset

In the future study, we plan to consider more complex neural network architectures. Other direction of research includes considering the neighborhood of the intersection to describe the system state.

REFERENCES

[1] C. Lim, K.-J. Kim, and P. P. Maglio, “Smart cities with big data: Reference models, challenges, and considerations,” *Cities*, vol. 82, pp. 86–99, Dec. 2018, doi: 10.1016/j.cities.2018.04.011.

[2] E. Ismagilova, L. Hughes, Y. K. Dwivedi, and K. R. Raman, “Smart cities: Advances in research—An information systems perspective,” *International Journal of Information Management*, vol. 47, pp. 88–100, Aug. 2019, doi: 10.1016/j.ijinfomgt.2019.01.004.

[3] A. Yumaganov, A. Agafonov, and V. Myasnikov, “Map Matching Algorithm Based on Dynamic Programming Approach,” in *2020 15th Conference on Computer Science and Information Systems (FedCSIS)*, Sep. 2020, pp. 563–566. doi: 10.15439/2020F139.

[4] A. A. Agafonov, “Short-Term Traffic Data Forecasting: A Deep Learning Approach,” *Optical Memory and Neural Networks*, vol. 30, no. 1, pp. 1–10, Jan. 2021, doi: 10.3103/S1060992X21010021.

[5] A. Adart, H. Mouncif, and M. Naïmi, “Vehicular ad-hoc network application for urban traffic management based on markov chains,” *International Arab Journal of Information Technology*, vol. 14, no. 4A Special Issue, pp. 624–631, 2017.

[6] Y. Li, E. Fadda, D. Manerba, R. Tadei, and O. Terzo, “Reinforcement Learning Algorithms for Online Single-Machine Scheduling,” in *2020 15th Conference on Computer Science and Information Systems (FedCSIS)*, Sep. 2020, pp. 277–283. doi: 10.15439/2020F100.

[7] B. N. Silva, M. Khan, and K. Han, “Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities,” *Sustainable Cities and Society*, vol. 38, pp. 697–713, Apr. 2018, doi: 10.1016/j.scs.2018.01.053.

[8] B. Xu, X. J. Ban, Y. Bian, J. Wang, and K. Li, “V2I based cooperation between traffic signal and approaching automated vehicles,” in *2017 IEEE Intelligent Vehicles Symposium (IV)*. Los Angeles, CA, USA: IEEE, Jun. 2017, pp. 1658–1664. doi: 10.1109/IVS.2017.7995947.

[9] C. Yu, Y. Feng, H. Liu, W. Ma, and X. Yang, “Integrated optimization of traffic signals and vehicle trajectories at isolated urban intersections,” *Transportation Research Part B: Methodological*, vol. 112, pp. 89–112, 2018, doi: 10.1016/j.trb.2018.04.007.

[10] H. Wei, G. Zheng, V. Gayah, and Z. Li, “A Survey on Traffic Signal Control Methods,” *arXiv:1904.08117 [cs, stat]*, Jan. 2020, arXiv: 1904.08117. [Online]. Available: http://arxiv.org/abs/1904.08117

[11] S. S. S. M. Qadri, M. A. Gökçe, and E. Öner, “State-of-art review of traffic signal control methods: challenges and opportunities,” *European Transport Research Review*, vol. 12, no. 1, p. 55, Dec. 2020, doi: 10.1186/s12544-020-00439-1.

[12] Q. Guo, L. Li, and X. (Jeff) Ban, “Urban traffic signal control with connected and automated vehicles: A survey,” *Transportation Research Part C: Emerging Technologies*, vol. 101, pp. 313–334, Apr. 2019, doi: 10.1016/j.trc.2019.01.026.

[13] M. Papageorgiou, C. Kiakaki, V. Dinopoulou, A. Kotsialos, and Yibing Wang, “Review of road traffic control strategies,” *Proceedings of the IEEE*, vol. 91, no. 12, pp. 2043–2067, Dec. 2003, doi: 10.1109/JPROC.2003.819610.

[14] R. Allsop, “Estimating the traffic capacity of a signalized road junction,” *Transportation Research*, vol. 6, no. 3, pp. 245–255, 1972, doi: 10.1016/0041-1647(72)90017-2.

[15] F. V. Webster, *Traffic Signal Settings*. H.M. Stationery Office, 1958.

[16] J. Little, M. Kelson, and N. Gartner, “MAXBAND: A Program for Setting Signals on Arteries and Triangular Networks,” *Transportation Research Record Journal of the Transportation Research Board*, vol. 795, pp. 40–46, Dec. 1981.

[17] M.-T. Li and A. Gan, “Signal timing optimization for oversaturated networks using TRANSYT-7F,” *Transportation Research Record*, no. 1683, pp. 118–126, 1999, doi: 10.3141/1683-15.

[18] P. Varaiya, “The Max-Pressure Controller for Arbitrary Networks of Signalized Intersections,” in *Advances in Dynamic Network Modeling in Complex Transportation Systems*, ser. Complex Networks and Dynamic Systems, S. V. Ukkusuri and K. Ozbay, Eds. New York, NY: Springer, 2013, pp. 27–66. doi: 10.1007/978-1-4614-6243-9_2.

[19] K. Stoilova and T. Stoilov, “Bi-level Optimization Application for Urban Traffic Management,” in *2020 15th Conference on Computer Science and Information Systems (FedCSIS)*, Sep. 2020, pp. 327–336. doi: 10.15439/2020F18.

[20] K.-L. Yau, J. Qadir, H. Khoo, M. Ling, and P. Komisarczuk, “A survey on Reinforcement learning models and algorithms for traffic signal control,” *ACM Computing Surveys*, vol. 50, no. 3, 2017, doi: 10.1145/3068287.

[21] M. Gregurić, M. Vujić, C. Alexopoulos, and M. Miletić, “Application of Deep Reinforcement Learning in Traffic Signal Control: An Overview and Impact of Open Traffic Data,” *Applied Sciences*, vol. 10, no. 11, p. 4011, Jun. 2020, doi: 10.3390/app10114011.

[22] P. Palos and A. Huszak, “Comparison of Q-Learning based Traffic Light Control Methods and Objective Functions,” in *2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. Split, Hvar, Croatia: IEEE, Sep. 2020, pp. 1–6. doi: 10.23919/SoftCOM50211.2020.9238290.

[23] H. Wei, G. Zheng, H. Yao, and Z. Li, “IntelliLight: A Reinforcement Learning Approach for Intelligent Traffic Light Control,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. London United Kingdom: ACM, Jul. 2018, pp. 2496–2505. doi: 10.1145/3219819.3220096.

[24] H. Wei, N. Xu, H. Zhang, G. Zheng, X. Zang, C. Chen, W. Zhang, Y. Zhu, K. Xu, and Z. Li, “CoLight: Learning Network-level Cooperation for Traffic Signal Control,” *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 1913–1922, Nov. 2019, arXiv: 1905.05717, doi: 10.1145/3357384.3357902.

[25] C. Chen, H. Wei, N. Xu, G. Zheng, M. Yang, Y. Xiong, K. Xu, and Z. Li, “Toward A Thousand Lights: Decentralized Deep Reinforcement Learning for Large-Scale Traffic Signal Control,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, pp. 3414–3421, Apr. 2020, doi: 10.1609/aaai.v34i04.5744.

[26] Y. Liu, L. Liu, and W.-P. Chen, “Intelligent Traffic Light Control Using Distributed Multi-agent Q Learning,” *arXiv:1711.10941 [cs]*, Nov. 2017, arXiv: 1711.10941. [Online]. Available: http://arxiv.org/abs/1711.10941

[27] Z. Li, H. Yu, G. Zhang, S. Dong, and C.-Z. Xu, “Network-wide traffic signal control optimization using a multi-agent deep reinforcement learning,” *Transportation Research Part C: Emerging Technologies*, vol. 125, p. 103059, Apr. 2021, doi: 10.1016/j.trc.2021.103059.

[28] J. Gu, Y. Fang, Z. Sheng, and P. Wen, “Double Deep Q-Network with a Dual-Agent for Traffic Signal Control,” *Applied Sciences*, vol. 10, no. 5, p. 1622, Feb. 2020, doi: 10.3390/app10051622.

[29] J. Zeng, J. Hu, and Y. Zhang, “Adaptive Traffic Signal Control with Deep Recurrent Q-learning,” in *2018 IEEE Intelligent Vehicles Symposium (IV)*. Changshu: IEEE, Jun. 2018, pp. 1215–1220. doi: 10.1109/IVS.2018.8500414.

[30] H. Hasselt, “Double Q-learning,” *Advances in Neural Information Processing Systems*, vol. 23, 2010.

[31] H. Zhang, S. Feng, C. Liu, Y. Ding, Y. Zhu, Z. Zhou, W. Zhang, Y. Yu, H. Jin, and Z. Li, “CityFlow: A Multi-Agent Reinforcement Learning Environment for Large Scale City Traffic Scenario,” *arXiv:1905.05217 [cs]*, May 2019, arXiv: 1905.05217. [Online]. Available: http://arxiv.org/abs/1905.05217, doi: 10.1145/3308558.3314139.

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 371–378
DOI: 10.15439/2021F120
ISSN 2300-5963 ACSIS, Vol. 25

The Virtual Emotion Loop: Towards Emotion-Driven Product Design via Virtual Reality

Davide Andreoletti, Luca Luceri, Achille Peternier, Tiziano Leidi and Silvia Giordano
University of Applied Sciences and Arts of Southern Switzerland, Lugano, Switzerland; name.surname@supsi.ch
Email: davide.andreoletti@supsi.ch

***Abstract*—Emotions play a significant role in product design for end-users. However, how to take emotions into account is not yet completely understood. We argue that this gap is due to a lack of methodological and technological frameworks for effective investigation of the elicitation conditions related to emotions and corresponding emotional responses of the users. Emotion-driven design should encompass a thorough assessment of users' emotional reactions in relation to certain elicitation conditions. By using Virtual Reality (VR) as mean to perform this investigation, we propose a novel methodological framework, referred to as the VR-Based Emotion-Elicitation-and-Recognition loop (VEE-loop), to close this gap.**

I. Introduction

The capability of a product (being it tangible or not, e.g., a service) to engage its users at the emotional level is considered by many the main factor behind its success [1]. In this respect, the authors of Ref. [2] claim that up to 95% of our buying decisions are not driven by rational arguments. Despite this, the fulfillment of functional requirements is generally the only objective in product design, with emotional aspects being often underestimated, if not even totally disregarded [3]. While this contradiction is commonly interpreted as a phenomenon of cultural inertia [3] (an argument that we also support), in this paper we give the following additional interpretation: there is no methodological and technological framework that helps designers to systematically experiment emotion elicitation conditions, as well as to assess the consequent users' emotions.

Emotion-driven design is the set of processes and methods used for developing products with the specific aim of evoking certain emotional responses. Following this paradigm, designers consider the final users' emotions since the very early stages of development, and not, as often done, by leaving them as an afterthought. Existing approaches aimed to foster emotion-driven design (e.g., [1]) are based on the idea that designers should familiarize themselves with all the aspects of emotions, ranging from their definition and elicitation conditions (e.g., in terms of products' sensory characteristics) and manifestations on people. Following a similar line of reasoning, in this paper we argue that emotion-driven design should be characterized by i) a systematic experimentation of various emotion elicitation conditions (e.g., the sensory qualities of the product and context of its usage) and by ii) the reliable measurement of users' emotional response, i.e., emotion recognition. Then, we also argue that emotional elicitation and emotion recognition should be iterated in a continuous loop until the emotion elicitation conditions can evoke the emotion intended by the designer. Specifically, this argument is supported by the fact that many existing works (e.g., [4], [5]) claim that iterative product design is an effective means to continuously improve a product and to adapt it to changes in customer demand. According to these methods, the opinions of several persons (e.g., of professionals, but also of the customers themselves) are considered to modify the product in a more informed and targeted way. This process is iterated until designers converge to a set of widely agreed product characteristics.

The scheme envisioned in our paper follows a similar iterative method and considers the emotions of users as the main implicit feedback that should drive the modification of product characteristics. Indeed, according to the proposed scheme, a designer makes her stylistic choice with the intention to evoke some emotional response (i.e., emotion elicitation phase); users' emotions are qualitatively measured (i.e., emotion recognition phase); the designer observes the emotions actually perceived by users and gains relevant insights to adjust her choices accordingly (i.e., loop phase). We are aware that the emotional reaction might be significantly affected by the number of iterations. Just as an example, boredom might be evoked more frequently after having done several experiments, regardless of the current emotion elicitation conditions. We discuss this and similar issues in Section VI. Please note that the proposed scheme can be implemented in the design phase (e.g., to validate the hypothesis that some feature triggers a specific emotional reaction) and, whenever possible, in the delivery phase as well (e.g., by dynamically adapting the characteristic of a service in real-time). While the proposed scheme can, in principle, be implemented using the most varied approaches, we believe that Virtual Reality (VR) provides the perfect controlled environment to turn our vision into a practical design instrument. Therefore, we refer to the proposed framework as the VEE-loop, i.e., the Virtual-Reality-Based Emotion-Elicitation-and-Recognition loop.

We identify several factors that, in our view, make VR the most suitable technology to implement the proposed scheme. First, among all the existing digital technologies, VR is the one that guarantees the most tangible experience across different domains. In fact, VR allows users to feel a sense of presence that makes emotion elicitation conditions quite similar to a real scenario [6], with the remarkable advantage of also enabling a flexible modification of the virtual scene experienced by the user, i.e., the Virtual Environment (VE). In addition, VR allows gathering a set of users-related data from which their emotions can be inferred (e.g., bio-feedback and behaviors). Note that most bio-feedback signals can be directly gathered through the Head Mounted Displays (HMDs) used by the VR system, coupled with external devices (e.g., wearables) when needed.

The VEE-loop has the potential to benefit a high number of application areas. For example, it can be used as a tool to perform a validation of the capability of a product to trigger specific emotions, before its actual production. Indeed, designers could obtain an emotional feedback from potential customers and tune the design accordingly. Given that this feedback is obtained before the actual product development, the risk of designing unsuccessful products is significantly reduced. In this respect, the immersion level provided by VR guarantees a higher fidelity of this emotional reaction with respect to other methods, while its flexibility allows testing a high number of products' characteristics. The VEE-loop can also improve services in which the knowledge of users' affective states is highly beneficial, but not always available (e.g., due to physical distancing measures imposed to handle the Covid-19 pandemic). In remote learning, for example, the emotional states of students can be monitored, and the virtual lecture dynamically changed (e.g., to induce calm in students or to draw their attention) [7].

Our work's main contributions are: (i) we present an extensive review and discussion on the importance of emotions in product design and delivery, (ii) we provide a formalization of iterative emotion-driven design approaches, and (iii) we propose a technological framework to realize this approach by means of VR. The structure of the paper, which reflects these contributions, is as follows. In Section II we elaborate on the importance of emotions in user-product interaction, we present the characteristics of emotion-driven product design, and we motivate the use of VR as the enabling technology to implement it. In Section III we describe the VEE-loop in detail, while in Section IV we show how the VEE-loop advances existing approaches. Section V is devoted to the presentation of the impact and of the application areas of the VEE-loop. Finally, in Section VI we discuss opportunities and challenges derived from the use of the VEE-loop.

II. Towards VR-based Emotion-Driven Product Design

This Section starts by elaborating on the importance of emotions in users-product interaction. Then, it introduces the paradigm of emotion-driven product design, highlighting its main characteristics. Finally, it motivates the use of VR as an enabling technology towards the realization of this paradigm.

A. Emotions in users-product interaction

Emotions are present in almost all human experiences, including the interaction between users and products. Based on the findings of previous work, the authors of Ref. [1] identify three main product characteristics that evoke emotions on their users, namely *appearance*, *functionality*, and *symbolic meaning*. As for the appearance, it is acknowledged that sensory qualities (e.g., shape and color) are associated with different emotional experiences. This concept is the basis of the *Kansei engineering model* [8], according to which there exists a causal relation between the attributes of a product and the emotional response of its users. For instance, warm colors are generally chosen to increase the arousal levels of evoked emotions [9]. In general, the functionalities of a product elicit positive emotions if they fulfill the needs of the users, and negative otherwise. For instance, a product that improves a situation that is perceived as frustrating and limiting (e.g., by enabling to gain space in small environments) is likely to evoke positive emotions. On the contrary, a product that is cumbersome and reduces the available space likely leads to frustration and annoyance. Then, the symbolic meaning of a product refers to its connection with a broader scheme of beliefs and values. In relation to the symbolic meaning of a product, the appraisal theory [10] states that emotions are triggered by the foretaste that users have when they evaluate the role of the product in their lives. For instance, a treadmill can evoke positive emotions in those who see it as a mean to get fitter, but negative ones in those worried by strain. Another example is the symbolic value given by the affinity of a product with a certain idea (e.g., a flag that represents a certain political view).

The importance of the symbolic meaning of products in evoking emotions is well expressed in the famous quote stated by Simon Sinek in one of the most viewed TED talk ever[1]: *people don't buy what you do, they buy why you do it* [11]. Indeed, the meaning that a person ascribes to a product is strongly correlated with her inner values, and their affinity with a company's mission and concerns. In relation to this, the *law of concern* formulated in [12] affirms that every emotion hides a personal concern and a disposition to prefer particular states of the world. This fact has led the authors of Ref. [13] to describe emotions as gateways to what people really care for, and entry points to uncover their underlying concerns. Along similar lines, the authors of Ref. [3] argue that understanding users at the emotional level allows having a deeper comprehension of their values, which is crucial to produce radical product innovations, while the sole understanding of users' functional needs yields only superficial and slight product modifications. Moreover, products capable to emotionally engage their users foster creative and innovative thinking [13], and benefit well-being [14]. Therefore, the capability of understanding and engaging users at the emotional level is crucial to design products that are appreciated and guarantee loyalty of customers in the long term (in this respect, note also that a clear and tight connection between a product and a specific emotion reinforces brand identification [3]).

B. Characteristics of emotion-driven product design

Given the major role of emotions in the relation between users and products, **emotion-driven design**, i.e., the realization of products with the deliberate intention to evoke specific

[1] https://www.ted.com/talks/simon_sinek_how_great_leaders_inspire_action?language=en

emotions [13], is rapidly becoming an important research area. Several frameworks have been recently proposed to help designers in the creation of products with emotional intentions. These frameworks (e.g., the Emotion-Driven Innovation paradigm [1]) share the idea that designers should be supported in the acquisition and practical exploitation of a solid *emotion knowledge*, which is defined in Ref. [13] as the explicit understanding of the physical manifestations of emotions and of their eliciting conditions. These frameworks are still not very employed [14]; arguably, this limited diffusion is mainly due to the lack of a technological layer to facilitate the study of emotion elicitation and recognition. In this paper, we claim that VR represents the most powerful medium to invert this tendency and consolidate the practice of emotion-driven design. In the following, we express our view on the characteristics that a framework for emotion-driven design should have, both concerning the study of the conditions of emotion elicitation and the recognition of emotions from their manifestation. Then, in subsection II-C, we provide arguments that support the idea of using VR as the basis to implement such a framework.

1) Emotion Recognition: The capability to disambiguate between different emotions (e.g., to understand the difference between frustration and annoyance) has been defined in Ref. [13] as emotional granularity and is regarded in Ref. [15] as a core advantage for the realization of emotion-driven products. Indeed, it is essential that designers understand the nuances of emotions beyond the simple positive versus negative distinction [14] (in this respect, just think that consumers may experience 25 different positive emotions when interacting with a product [13]).

The most straightforward approach to understand users' emotions (and, more in general, to understand their perception of a product [16]) is by direct communication. However, people are generally not aware of their emotions, nor can they properly verbalize and communicate them. Hence, traditional investigation tools (e.g., surveys and interviews) cannot effectively capture users' emotions. Moreover, tools based on self-reports require users to interrupt their activity, which in turn may hinder the validity of their records [13]. Therefore, emotion recognition techniques, i.e., the qualitative measurement of emotions from their manifestations, seems to be the most viable alternative.

Affective computing refers to a set of technologies and strategies developed to automatize emotion recognition, generally exploiting machine learning algorithms that infer the emotions a person most likely perceives from his/her bio-feedback (e.g., facial expression, blood pressure, movements, etc.). The fact that only bio-feedback are considered is, in our view, a severe limitation of the traditional approach. For instance, behaviors of users, which are actually part of the manifestations of emotions [14], are currently disregarded. Indeed, emotions are complex phenomena that are better understood if studied holistically [14]. In user-product interaction, this holistic investigation would require, for example, the correlation of the sensory and symbolic characteristics of the product with the bio-feedback of the users, as well as with her behaviors (e.g., which action users take after using a product) and with the context in which the product is used [13]. In particular, the possibility to correlate users' emotions with contextual factors has been considered in Ref. [14] as a way to better understand their concerns and inner values. Therefore, we believe that a framework for emotion-driven design should allow probing users' emotions considering as many aspects of their manifestations as possible (e.g., bio-feedback, context, behavior, etc.).

2) Emotion Elicitation: The task of identifying the conditions that elicit certain emotions is inherently challenging. Indeed, while it is acknowledged that certain products' characteristics induce similar emotional reactions on most of their users [9], emotions are generally subjective experiences. In other words, the relation between certain types of stimuli and emotions is neither deterministic nor constant, as it can change over time even for the same person [13]. In addition, elicitation conditions are extremely complex, as they depend on the interaction of various factors, such as product's characteristics and context of usage.

Considering this, we believe that a framework for emotion-driven design should favor the validation of emotion elicitation conditions both on a single-user and on a large-scale basis. As for the former, this framework would help designers to understand a customer at the emotional level, and in turn uncover her latent concerns and desires [3]. In addition, this framework would enable the design of products performed cooperatively by designers and customers (note that product co-design is considered more attractive than designer-only and customer-only design [3]). As for the latter, the framework should allow to easily validate the effectiveness of different combinations of elicitation conditions (e.g., sensory qualities of the product and environment in which the product is used) in evoking certain emotions.

3) Loop of Emotion Recognition and Elicitation: Finally, our envisioned framework follows the iterative approach of existing design strategies (e.g., [4]). In fact, users' emotions should be continuously tracked and provided to designers as a feedback of the suitability of the chosen elicitation conditions. Elicitation conditions can then be modified accordingly, in a more informed manner. This operation can then be repeated until the expected emotional reaction is achieved, which gives rise to the **Emotion-Recognition-Emotion-Elicitation Loop**. This scheme can be applied, for instance, to validate the effectiveness of experimental emotion elicitation conditions, as well as to better understand the factors that triggered some particular emotions. In the following subsection, we provide arguments that support our choice of VR as the most suitable candidate technology to implement this scheme, which we then refer to as the **VR-based Emotion-Recognition-Emotion-Elicitation Loop**, or simply the VEE-loop. A representation of the VEE-loop is depicted in Fig. 1.

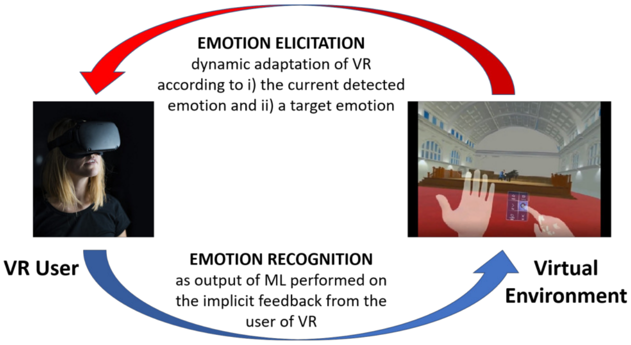

EMOTION ELICITATION
dynamic adaptation of VR according to i) the current detected emotion and ii) a target emotion
VR User
EMOTION RECOGNITION
as output of ML performed on the implicit feedback from the user of VR
Virtual Environment

Fig. 1: High-Level Representation of the VEE loop

C. VR as enabling technology

Several factors make VR the most natural, direct, and suitable technology to implement the framework described so far. First, pure VR (i.e., a user interacting with entirely synthetic, computer-generated VEs) allows creating completely modifiable, dynamic experiences. Unlike augmented and mixed reality, which is limited and linked to the surrounding physical elements, pure VR can be easily distributed online, experienced everywhere, replayed at will, and its content regularly updated. The immersion provided by VR also amplifies emotional reactions [6], [17], [18], which helps both in the emotion recognition and elicitation phases compared to other less effective means to put a user in a given simulated situation. In addition, the retention rate of learning and training provided via VR is increased when compared to more conventional media [19]. The recent uptake in VR adoption as well as increased availability of affordable VR headsets provide a much lower entry-point to the technology, which is now considered a commodity, off-the-shelf option no longer limited to research laboratories or professional contexts. Thanks to this evolution, which also significantly increased the quality of modern VR compared to the state of the art of just few years ago, a significantly larger user-base can now be targeted by VR-enabled solutions.

Then, modern VR equipment already embeds sensors that are critical for inferring the user's emotional state (and its evolution) during the virtual experience. Since body tracking is a central requirement of VR, most of the recent HMDs are capable of tracking user's head and hands position in real-time and at high frequency, while some models include eye tracking as well. These sensors can be used for the proper positioning of the user within the VE (e.g., to update the viewpoint and stereoscopic rendering parameters), as well as to precisely determine what the user is currently looking at, but also to derive a series of additional metrics such as heartbeat and respiratory rate [20]. Next-generation HMDs will directly embed dedicated sensors for monitoring such states (like the HP Reverb G2 Omnicept).

This constant source of information can be used to acquire data that previously required to equip the user with a cumbersome set of devices and to prepare the environment for different levels of motion tracking (from a simple Microsoft Kinect to professional-grade systems such as the Vicon). Most of these capabilities are now integrated into one single device that provides all the ingredients for building an emotion recognition and elicitation system under wearable and affordable constraints. Nevertheless, HMDs can still be coupled with additional monitoring devices to increase the amount, types, and accuracy of users' bio-feedback signals for this task (e.g., by combining the full-body tracking provided by the Microsoft Kinect with the head and hands positions returned by the headset). Moreover, tools have been recently proposed to enable HMDs tracking the movements of the face[2]. In addition, VR enables simulating the context in which a user acts and analyzing his/her behaviors in relation to this. Let us note that, since emotions are observable from behaviors as well, this property of VR has the potential to revolutionize the research in emotion recognition and to increase the effectiveness of this task.

III. The VEE-Loop

The VEE-loop is the realization, by means of VR technologies, of the Emotion-Recognition-Emotion-Elicitation loop described in subsection II-B3. More specifically, the VEE-loop is implemented by continuously monitoring the affective states of users and by adapting the VE accordingly. A modification of the VE is performed, for instance, to induce a transition from the current to the desired affective state (e.g., from fear to calm), or to evaluate the effectiveness of some emotion elicitation conditions. The VEE-loop is composed of a module for Emotion Recognition (ER module) and one for Emotion Elicitation (EE module). A detailed representation of the VEE-loop architecture is depicted in Fig. 2. From this figure, it is possible to observe that the ER module infers the emotion most likely perceived by the user by elaborating information such user's bio-feedback and the interaction between user and VE (e.g., her behavior or the attention she pays to a particular virtual object). The emotion detected by the ER module and the emotion that the designer aims to evoke are then given in input to the EE module, which is responsible for dynamically changing the VE. We further articulate the ER and EE components in the next subsections.

A. Emotion Recognition Module

The ER module is responsible for inferring, from a set of multi-modal signals, the emotion that the user most likely perceives. We identify two main categories of these data: i) users' bio-feedback signals and ii) user-VE interactions. As far as user's bio-feedback are concerned (e.g., movements, vital parameters, etc.), we note that their acquisition can be performed directly with the HMD (e.g., by tracking head and eye movement), as well as with other supporting tools that do not hinder VR experience (e.g., wearable devices). Then, we also argue that users' emotions are strongly correlated with her interaction with the VE, e.g., fascination is observable from

[2] https://uploadvr.com/htc-facial-tracker-quest-index/

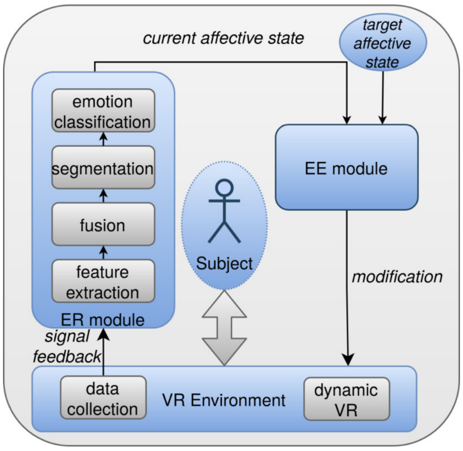


Fig. 2: Architecture of the proposed VEE-Loop

the level of attention and the time spent using some object [14]. To our knowledge, however, the problem of modeling the interaction between user and VE has never been considered in relation to the task of emotion recognition. In particular, as users' behaviors are integral aspects of their emotional status, it is essential to model the behavior of the users within the VE (e.g., which actions are taken, which types of objects are used, etc.).

More formally, the Emotion Recognition module estimates the emotional states and the instants of transitions between successive emotions, i.e., $\hat{\mathcal{E}}\,(t)$ and the set $\left\{t^{(\star)}|\hat{\mathcal{E}}\left(t^{(\star)}-dt\right) \neq \hat{\mathcal{E}}\left(t^{(\star)}+dt\right)\right\}$, where $\hat{\mathcal{E}}\left(t^{(\star)}\right)$ is the estimated emotional state at time $t^{(\star)}$ and dt is the infinitesimal delta of time. For the processing of bio-feedback and contextual data, we envision an architecture composed of the following four layers:

- Feature Extraction: a set of suitable features has to be defined to capture the properties of users' bio-feedback and contextual data that are beneficial for the emotion recognition task. A quite established literature can help in the definition of features to represent a high number of bio-feedback, both handcrafted and automatically learned [21] (e.g., acceleration of joints for body's movements and spectrogram for voices, or learned with a deep learning approach). The definition of features to represent contextual data (e.g., interaction of the user with the VE), instead, requires a more pioneering attitude. We argue that existing features used to model the level of attention and engagement can help to define some new more emotion-oriented features.
- Fusion: this layer is meant to combine data, features and algorithms to maximally exploit the information contained in the users' data to increase the generalization of the ER module; a research challenge here is to combine data of different domains (e.g., voice, heart rate and interaction with the VE) that have radically different properties, such as acquisition frequency and temporal dynamic.
- Segmentation: please note that, during the use of VR, users' emotions may change over time. This layer performs data segmentation, i.e., it segments the stream into portions of signals that are coherent with respect to the particular emotion they carry. This is a remarkable difference with respect to existing studies on emotion recognition, which generally assume that observed data is associated with a single emotional state.
- Emotion Classification: finally, this layer infers the emotion that the obtained segmented data most likely carries. Note that emotions can be represented either using a set of classes (e.g., joy, fear, etc.), or using a dimensional approach (e.g., arousal, valence, dominance [22]). Based on the type of representation that is chosen, this layer performs a supervised learning task, either a classification or a regression.

B. *Emotion Elicitation Module*

The Emotion Elicitation (EE) module outputs a modified VE based on the following input: i) a representation of the current VE, ii) the emotion detected by the ER module and the iii) the emotion that designer aims to evoke. Firstly, the EE module computes a measure of distance between the emotion intended by the designer and that recognized by the ER module. Then, based on this distance, the properties of the current VE are appropriately modified (e.g., the color of a given virtual object is changed).

The main research questions that are still pending here are how to measure the distance between emotions, and how to modify the VE accordingly. We note that, in order to compute this distance, emotions should be better described using a dimensional representation (e.g., in the valence/arousal plane), which allows quantifying the difference among them. Then, the difficulty in modifying a content to elicit emotions is a well-known problem, in particular when advanced interactive media, such as VR, are considered [6]. In our view, the first step to tackle this problem is the definition of a representation of the VE that includes, for instance, positions, semantic (i.e., functional role) and the sensory qualities (e.g., shape and size) of the most salient virtual objects. The second step is the definition of a model that, as a function of the detected emotion, its distance with the target emotion, and the representation of the current VE, returns an indication on how the VE should be modified. For example, if the target emotion is joy, but the one detected by the ER module is sadness, the colors of the objects could be tuned to be warmer, given that warmer colors are usually associated with joy. In our view, this task is still very complex to be automatized, and would require the manual tuning of VE's characteristics. However, in case the VEE-loop became a tool of common use, it would function also as a tool for data collection. Specifically, the tuning choices of designers, along with the emotional reactions of users, might be collected and then used to train automatic systems (e.g., machine learning algorithms).

IV. Related Work

Early prototypes of the VEE-loop are present in the literature. For instance, in [23] an architecture to perform users' emotion-driven generation of a VE is proposed and validated in the context of mental health treatment. Such architecture is designed to detect users' emotions from the analysis of multiple types of bio-feedback (similarly to our ER module) and, accordingly, to generate a VE to stabilize them, e.g., to induce calm (similarly to our EE module). Whilst this existing approach is quite similar, in principle, to our idea of VEE-loop, it has the main drawback of not considering complex models of interaction between users and VE that, in our view, are essential to realize an instrument suitable for emotion-driven product design (for instance, in Ref. [23] the generated VE is a simple maze). Another work that investigates the use of VR as a tool to perform ER and EE can be found in Ref. [24], where ER is performed using a very simple machine learning algorithm that works on users' electroencephalograms, while EE is implemented using static VEs. Instead, in order to be a suitable tool for emotion-driven product design, the VEE-loop will consider a vast array of heterogeneous bio-feedback, complex models of interaction between users and the VE, dynamic VEs and more advanced machine learning algorithms.

Most of the research on ER is done on single-mode and standalone data (see the recent survey [25]), which carry acted and exaggerated emotions. Instead, the proposed framework allows considering streams of multi-mode data (which introduce the challenge of identifying the onset and end of emotions) and exploiting the sense of presence typical of VR experience to induce (and then, recognize) more spontaneous emotions [26].

Various works (e.g., [4], [5]) witness how iterative design is a consolidated practice. All these iterative schemes share the idea that product design should be done by incrementally modifying the characteristic of a product in response to the perception of various persons (e.g., designers and customers). Ref. [27] represents an attempt to partially automatize this iterative process by employing a genetic algorithm that suggests how to modify a product based on current products' characteristics and by users' perception.

Refs. [1] and [14] describe methodological frameworks that designers can follow to develop products with emotional intentions. However, as far as we know, we are the first to propose a technological solution that can help to perform emotion-driven and iterative product design. We also remark that the proposed VEE-loop can also be used to dynamically modify virtual products based on users' emotions (e.g., a service of remote schooling). In this respect, similar previous work (that, however, do not make use of VR technologies) are [28]–[30] and [9], which propose systems for emotion-driven recommendation and advertising, respectively. Similarly, a digital system that adapts the characteristics of a service based on users' emotions is proposed in [31]. Then, Ref. [32] proposes a gaming framework that changes the characteristics of the game based on users' emotions. Finally, Ref. [33] describes the realization of a smart office in which sensory features (e.g., light in the office) and tasks assigned to users are changed to regulate their emotions.

V. Impact and Application

In this Section, we describe the main practical applications of the VEE-loop, as well as its potential impact across several areas.

A. *Areas of application*

The VEE-loop is a versatile tool that opens the doors to a wide spectrum of applications. In particular, we identify the following three main areas of applications: 1) product design, 2) virtual service delivery and 3) research in emotion recognition and elicitation.

In product design, the VEE-loop can be used to validate the capability of a product to fulfill its emotional requirements (i.e., to check the consistency between intended and perceived emotions) before the actual and expensive tangible production. In fact, by exploiting the sense of presence given by VR, the emotions experienced by the users are guaranteed to be as much similar as possible to the real ones. Hence, users can try the virtual counterparts of the products under development and provide the designers with implicit feedback about the goodness of their functional and stylistic choices. Note also that, the VEE-loop being a digital tool, this validation can easily involve a higher number of users with respect to traditional on-site experiments, therefore increasing their validity. Besides improving the product, the received feedback also helps designers to better study their customers at the emotional level and to understand which factors reinforce brand identification [34], [35].

Moreover, services that are delivered using digital channels can benefit from the use of the VEE-loop. We refer, in particular, to services in the education field, where having the information on users' emotional states is highly beneficial [7], but unavailable for some reasons (e.g., in remote schooling during the Covid-19 pandemic), or in human-machine interaction, where the required equipment is too expensive or dangerous (e.g., in the training of practitioners in industry). In this type of services, which are delivered in real-time, the VEE-loop is implemented to adapt the VE to the current emotional status of the users, e.g., to calm them down when they are anxious. This can be done by modifying the sensory qualities of the virtual objects (as also done in product design), as well as by adapting the learning tasks to enhance users' experience. Moreover, the VEE-loop can facilitate the transition towards an increasingly-digitized society, where a number of services can be modified according to users' emotions. In theatrical exhibitions, for instance, the VEE-loop can be exploited to better understand the relation between emotions and acting.

Finally, the VEE-loop has the potential to advance the state of the art on the growing fields of emotion recognition and elicitation. As for the former, the VEE-loop can help to enhance current models for emotion recognition, e.g., by including the context embodied in the VE and users' behaviors.

As for the latter, the VEE-loop can be exploited in many different research areas to better study the effectiveness of elicitation conditions (e.g., in marketing, interior design, etc.).

B. Potential Impact

In light of the numerous possible applications described before, our proposed solution can potentially benefit various dimensions of our society, as detailed in the following.

a) Economical Impact: The VEE loop finds applications in a countless number of industrial sectors, while providing potential economical advantages both in the production and in the marketing phases. For example, it can be used by experts in advertising to understand what reinforces unique brand association, or by designers to evaluate users' emotional response to the characteristics of a product before its tangible development. This allows designers to take more informed decisions, therefore, reducing the risks (and associated costs) of creating unsuccessful products and services.

b) Social Impact: The VEE loop can help to deliver more empathetic services using VR, therefore bringing a high social impact across many different areas (e.g., remote schooling). Potential applications can also target the treatment of pathologies characterized by disorders on the emotional sphere [23] and collective training in emergency situations.

c) Environmental Impact: The VEE loop integrates emotional aspects into services delivered remotely, therefore increasing their adoption. This has the potential of improving remote working and practices, thus, limiting unnecessary travels, and, in turn, reducing the emissions produced by means of transport.

d) Research Impact: Our vision contributes to the research on ER and EE, and provides a tool that researchers can readily use in the studies relative to these fields. The VEE loop is a novel and timely solution that can be a potential cornerstone in many different projects (from the research-oriented to the more applicative ones), therefore enabling cross-fertilization between academy and industry.

e) Cultural Impact: The VEE loop enables avant-garde cultural events delivered with VR. For instance, stylistic choices of a cultural event (e.g., in theatrical representations) can be modified according to the emotional response of the audience (even if attending remotely) in real-time and in an economically-sustainable manner. This asset can find application in several cultural scenarios, e.g., theatre, virtual city trip and virtual museum tours.

VI. Discussion

In this Section, we summarize the main characteristics of the VEE-loop, and we discuss corresponding opportunities and challenges. In particular, we discuss various issues that need to be properly tackled before the proposed VEE-loop can be effectively employed as a tool to perform emotion-driven design.

Our most important claim is that the VEE-loop can help designers to develop products capable of inducing some specific emotion on their users. The flexible content adaptation and sense of immersion guaranteed by VR are just a couple of reasons that support this claim. In fact, VR allows experimenting with many virtual products' characteristics (e.g., shape, color and position), which can evoke emotions similarly to their real counterpart. However, the right emotion elicitation conditions are likely to be found after a number of iterations of the VEE-loop. The main problem of this iterative approach is that the emotions that users perceive are not only influenced by the current virtual content, but also by the number of iterations itself (e.g., stress and boredom might arise after a long session of experiments). This issue needs to be correctly tackled to not hinder the validity of the performed experiments. When the VEE-loop is used in the design phase, a possible approach consists in limiting the duration of the experiment to a certain amount of time, and to validate the effectiveness of emotion elicitation conditions through statistical analysis (e.g., which emotions have been perceived by most of the users involved in the experiments). As for experiments done to understand the emotional reaction of a specific user, instead, a possible strategy is to alternate VEs that carry an emotional content with VEs that are emotionally neutral, so to bring the user back to her normal conditions. More generally, VR allows simulating the context in which a product is used, e.g., the surrounding environment and the after-use experience. In light of this, the VEE-loop becomes even a more powerful design instrument, as designers are free to experiment a higher number of emotion elicitation conditions, and emotion recognition exploits a richer set of observations, e.g., both bio-feedback signals and users' behavior. In our view, future research should address two main open points: i) how to define suitable experiments to gain relevant insights from users' behaviors, which are an essential aspect of their emotions (e.g., how users interact with the VE, which elements they observe, etc.) and ii) how to automatize the dynamic modification of the VE.

Then, the high level of immersion guaranteed by VR lead users to perceive more spontaneous emotions that, for this reason, are a more valuable feedback for the designers, but also more difficult to identify. Indeed, most of the research on emotion recognition is based on the analysis of emotions that are voluntarily exaggerated and that, for this reason, are also easier to recognize. Finally, emotions must be estimated from a stream of signals and not, as generally done in previous work, from data that are assumed to carry a single emotion. Therefore, a segmentation process is required to identify the instants of transitions between two different affective states, before their actual classification. To our knowledge, the problem of segmentation is extensively considered in the action recognition task, but quite unexplored in the emotion recognition one.

VII. Conclusions

In this paper, we have identified the iteration of emotion elicitation and recognition phases as a desirable property of an emotion-driven design strategy. We have then provided arguments to support the idea of using VR to realize this iteration, that we have referred to as the Virtual-Reality-Based Emotion-Elicitation-and-Recognition loop (VEE-loop). In brief, VR allows creating virtual yet very realistic environments that designers can flexibly modify to induce specific emotional reactions. Indeed, VR inherits all the benefits of digital technologies (e.g., flexible and controlled content modification) without sacrificing the realism of the experience. All these aspects render the VEE-loop a promising methodological and technological framework that can benefit many areas such as product design, virtual service delivery, and the research in emotion recognition and elicitation in general. The VEE-loop represents a promising methodological and technological framework that can be exploited to effectively design emotion-driven products, provided that several issues (both at the technological and methodological sides) are properly handled.

Acknowledgement

This research was funded by the EU EIT Manufacturing initiative through the V-Machina project, and by the Innosuisse innovation project VR+4CAD. Many thanks to Bart Kevelham for his fruitful comments.

References

[1] T. Alaniz and S. Biazzo, "Emotional design: the development of a process to envision emotion-centric new product ideas," *Procedia Computer Science*, vol. 158, pp. 474–484, 2019.

[2] G. Zaltman, "The subconscious mind of the consumer (and how to reach it)," *Harvard Business School. Working Knowledge. Obtenido de http://hbswk. hbs. edu/item/3246. html*, 2003.

[3] C. Wrigley and K. Straker, *Affected: Emotionally engaging customers in the digital age*. John Wiley & Sons, 2019.

[4] R. Dou, Y. Zhang, and G. Nan, "Iterative product design through group opinion evolution," *International Journal of Production Research*, vol. 55, no. 13, pp. 3886–3905, 2017.

[5] P. B. Luh, F. Liu, and B. Moser, "Scheduling of design projects with uncertain number of iterations," *European Journal of Operational Research*, vol. 113, no. 3, pp. 575–592, 1999.

[6] G. Riva, F. Mantovani, C. S. Capideville, A. Preziosa, F. Morganti, D. Villani, A. Gaggioli, C. Botella, and M. Alcañiz, "Affective interactions using virtual reality: the link between presence and emotions," *CyberPsychology & Behavior*, vol. 10, no. 1, pp. 45–56, 2007.

[7] A. Ortigosa, J. M. Martín, and R. M. Carro, "Sentiment analysis in facebook and its application to e-learning," *Computers in human behavior*, vol. 31, pp. 527–541, 2014.

[8] M. Nagamachi, "Kansei engineering: a new ergonomic consumer-oriented technology for product development," *International Journal of industrial ergonomics*, vol. 15, no. 1, pp. 3–11, 1995.

[9] Y. Liu, O. Sourina, and M. R. Hafiyyandi, "Eeg-based emotion-adaptive advertising," in *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*. IEEE, 2013, pp. 843–848.

[10] P. Desmet, "Designing emotions," 2002.

[11] S. Sinek, *Start with why: How great leaders inspire everyone to take action*. Penguin, 2009.

[12] N. H. Frijda *et al.*, *The emotions*. Cambridge University Press, 1986.

[13] P. M. Desmet, S. F. Fokkinga, D. Ozkaramanli, and J. Yoon, "Emotion-driven product design," in *Emotion Measurement*. Elsevier, 2016, pp. 405–426.

[14] C. Kim, J. Yoon, P. Desmet, and A. Pohlmeyer, "Designing for positive emotions: Issues and emerging research directions," 2021.

[15] J. Yoon, A. E. Pohlmeyer, and P. Desmet, "When'feeling good'is not good enough: Seven key opportunities for emotional granularity in product development," *International Journal of Design*, vol. 10, no. 3, pp. 1–15, 2016.

[16] E. Sauerwein, F. Bailom, K. Matzler, and H. H. Hinterhuber, "The kano model: How to delight your customers," in *International working seminar on production economics*, vol. 1, no. 4, 1996, pp. 313–327.

[17] J. Diemer, G. W. Alpers, H. M. Peperkorn, Y. Shiban, and A. Mühlberger, "The impact of perception and presence on emotional reactions: a review of research in virtual reality," *Frontiers in Psychology*, vol. 6, p. 26, 2015. [Online]. Available: https://www.frontiersin.org/article/10.3389/fpsyg.2015.00026

[18] J. Marín-Morales, J. L. Higuera-Trujillo, A. Greco, J. Guixeres, C. Llinares, C. Gentili, E. P. Scilingo, M. Alcañiz, and G. Valenza, "Real vs. immersive-virtual emotional experience: Analysis of psycho-physiological patterns in a free exploration of an art museum," *PLOS ONE*, vol. 14, no. 10, pp. 1–24, 10 2019. [Online]. Available: https://doi.org/10.1371/journal.pone.0223881

[19] S. K. Babu, S. Krishna, U. R., and R. R. Bhavani, "Virtual reality learning environments for vocational education: A comparison study with conventional instructional media on knowledge retention." in *2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT)*, 2018, pp. 385–389.

[20] C. Floris, S. Solbiati, F. Landreani, G. Damato, B. Lenzi, V. Megale, and E. G. Caiani, "Feasibility of heart rate and respiratory rate estimation by inertial sensors embedded in a virtual reality headset," *Sensors*, vol. 20, no. 24, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/24/7168

[21] M. Buccoli, M. Zanoni, A. Sarti, S. Tubaro, and D. Andreoletti, "Unsupervised feature learning for music structural analysis," in *2016 24th European Signal Processing Conference (EUSIPCO)*. IEEE, 2016, pp. 993–997.

[22] A. Mehrabian, "Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament," *Current Psychology*, vol. 14, no. 4, pp. 261–292, 1996.

[23] S. B. i Badia, L. V. Quintero, M. S. Cameirão, A. Chirico, S. Triberti, P. Cipresso, and A. Gaggioli, "Toward emotionally adaptive virtual reality for mental health applications," *IEEE journal of biomedical and health informatics*, vol. 23, no. 5, pp. 1877–1887, 2018.

[24] J. Marín Morales, "Modelling human emotions using immersive virtual reality, physiological signals and behavioural responses," Ph.D. dissertation, 2020.

[25] A. Saxena, A. Khanna, and D. Gupta, "Emotion recognition and detection methods: A comprehensive survey," *Journal of Artificial Intelligence and Systems*, vol. 2, no. 1, pp. 53–79, 2020.

[26] S. Susindar, M. Sadeghi, L. Huntington, A. Singer, and T. K. Ferris, "The feeling is real: Emotion elicitation in virtual reality," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 63, no. 1. SAGE Publications Sage CA: Los Angeles, CA, 2019, pp. 252–256.

[27] K. Fung, C. K. Kwong, K. W. M. Siu, and K. M. Yu, "A multi-objective genetic algorithm approach to rule mining for affective product design," *Expert Systems with Applications*, vol. 39, no. 8, pp. 7411–7419, 2012.

[28] M. Polignano, F. Narducci, M. de Gemmis, and G. Semeraro, "Towards emotion-aware recommender systems: an affective coherence model based on emotion-driven behaviors," *Expert Systems with Applications*, vol. 170, p. 114382, 2021.

[29] M. B. Mariappan, M. Suk, and B. Prabhakaran, "Facefetch: A user emotion driven multimedia content recommendation system based on facial expression recognition," in *2012 IEEE International Symposium on Multimedia*. IEEE, 2012, pp. 84–87.

[30] N. Sindhu, S. Jerritta, and R. Anjali, "Emotion driven mood enhancing multimedia recommendation system using physiological signal," in *IOP Conference Series: Materials Science and Engineering*, vol. 1070, no. 1. IOP Publishing, 2021, p. 012070.

[31] N. Condori-Fernandez, "Happyness: an emotion-aware qos assurance framework for enhancing user experience," in *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*. IEEE, 2017, pp. 235–237.

[32] M. S. Hossain, G. Muhammad, B. Song, M. M. Hassan, A. Alelaiwi, and A. Alamri, "Audio–visual emotion-aware cloud gaming framework," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 12, pp. 2105–2118, 2015.

[33] S. Munoz, O. Araque, J. F. Sánchez-Rada, and C. A. Iglesias, "An emotion aware task automation architecture based on semantic technologies for smart offices," *Sensors*, vol. 18, no. 5, p. 1499, 2018.

[34] V. De Luca, "Emotions-based interactions: Design challenges for increasing well-being," 2016.

[35] V. a. De Luca, "Oltre l'interfaccia: emozioni e design dell'interazione per il benessere," *MD Journal*, vol. 1, no. 1, pp. 106–119, 2016.

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 379–390
DOI: 10.15439/2021F26
ISSN 2300-5963 ACSIS, Vol. 25

Artificial Intelligence Project Success Factors: Moral Decision-Making with Algorithms

Gloria J. Miller
Managing Consultant
maxmetrics
Heidelberg, Germany
https://orcid.org/0000-0003-2603-0980

***Abstract*—The algorithms implemented through artificial intelligence (AI) and big data projects are used in life-and-death situations. While research exists to address varying aspects of moral decision-making with algorithms, the definition of project success is not readily available. Nevertheless, researchers place the burden of responsibility for ethical decisions from AI systems on the system developers. Using a systematic literature review, this research identified 71 AI project success factors in 14 groups related to moral decision-making with algorithms. It contributes to project management literature, specifically for AI projects. Project managers and sponsors can use the results during project planning and execution.**

***Index Terms*—artificial intelligence, algorithms, moral decision making, critical success factors, project management**

I Introduction

ALGORITHMIC decision-making is replacing or augmenting human decision-making across many industries and functions [1, 2]. The decisions range from trivial to life and death. For example, marketing decisions are insignificant compared to legal decisions that may result in incarceration for defendants or the loss of life for health decisions affecting patients. An “algorithm is a defined, repeatable process and outcome based on data, processes, and assumptions” [3]. Algorithms are usually the result of artificial intelligence (AI) or big data projects. It is anticipated that AI will significantly impact society, generating productivity and efficiency gains and changing the way of work [4]. Given the considerable impact on individuals, society, and the environment, understanding the success factors in AI projects is critical.

Sponsoring organizations invest in AI projects expecting them to deliver measurable, meaningful benefits such as revenue or productivity gains [5]. The benefits of AI projects are usually realized long after the projects are completed and the algorithms are put into use. However, the on-time and cost limits of the task or the goal orientation of projects create the risk that the interests of significant stakeholders may not be considered. Thus, the short-term project objectives compared to the long-term social and environmental consequences raise essential questions about the definition of project success.

The decisions or results of the algorithm are what affect the individual and society. The development of large-scale AI models is what affects the environment. Thus, the definition of project success from the public’s perspective should be based on quality, morality, or fairness. The technology view of moral decision-making with AI does not consider non-technical stakeholders, e.g., operators and the public [6]. Manders-Huits [7] explains that the notion of consequences and the level of autonomy of action are preconditions or considerations for moral responsibility, arguing that the burden of responsibility for moral decisions is on the system designers' shoulders. Martin [8] makes a similar argument stating, “Developers are those most capable of enacting change in the design and are sometimes the only individuals in a position to change the algorithm.” Thus, while research exists to address varying aspects of moral decision-making with algorithms, the definition of project success is not readily available.

The project management literature clarifies that many project stakeholders measure success at different periods and do not share views on the success [9-11]. While the literature acknowledges the importance of client consultation and client acceptances as critical success factors, the public is not foreseen in an active project role. Furthermore, morality is not considered an independent project objective. However, [12] argues that managers should serve legitimate stakeholders' legal and moral interests.

This research uses a systematic review of the literature to answer a novel question regarding the success factors in AI projects: *what are the project success factors for moral decision-making with algorithms*? It closes the gap on a lack of literature that translates the AI ethical principles into practice [13]. Furthermore, the management of AI projects is hardly covered in the project management literature. This research contributes to the literature on success factors, specifically for AI projects. The paper is structured to provide a literature review, description of the methodology, findings, discussion of the research questions, and conclusions.

IEEE Catalog Number: CFP2185N-ART ©2021, PTI

II. Literature Review

A. Project success factors

Projects are temporary endeavors with their termination planned from the beginning. Thus, the project objectives and success criteria should be agreed upon with the stakeholders before starting a project [14, 15]. The long-term orientation needed to consider passive stakeholders contradicts the temporality of projects unless the long-term perspective is considered in the project objectives, business case, and investments [15]. Project success refers to the project delivering its expected output and achieving its intended objective. In contrast, project efficiency refers to the project management success regarding time, costs, and quality—the iron triangle [9, 14]. Success criteria and success factors are the dimensions for the stakeholder perceptions of project success [9, 14]. The criteria measure success while factors identify the circumstances, conditions, and events for reaching the objectives. The efficiency of the project can be measured when the outputs are produced. In contrast, project benefits and organizational performance impacts can be measured after the project outputs have been put into operations.

Several project management critical success factor models exist, the Pinto and Slevin [16] model being the most referenced [14]. It defines ten success factors under the control of the project team and four factors that influence project success but are not under the project team's control. Rather than identifying specific project success factors, [17] identified four groups of interrelated factors that could be analyzed across any type of project. While each model considers internal and external factors, their scope is bound by the project objectives. Customer consultation and acceptance are success factors; however, the public is not foreseen in that role. Using the framework from [16], [18] identified ethical knowledge as a specialized skill needed by the project personnel, questioned the role of moral decision-making in extreme situations, and identified ethical concerns as a risk to manage. However, the study does not explicitly address morality as a project objective.

The [9] project success model examines how stakeholders perceive success after completing the project. It was the first model to look at success outside the typical project life cycle and simultaneously consider multiple stakeholders [11]. The model defines the project results at multiple timescales: project outputs at the end of the project, outcomes months after the end, and impacts years after the end. It considers eight stakeholder groups: investors or owners, consumers, operators/users, project executive or project sponsor, project manager and project team, senior supplier, other suppliers, and public. Each stakeholder group and timescale provide success indicators, such as cost, features, performance, benefits, documentation, training, retention, well-being, learning, profit, new capabilities, future work, and new competence.

B. AI projects

AI encompasses multiple disciplines or branches within computer science. Natural Language Processing (NLP) covers making the computer understand, process, and manipulate human language [19]. Pattern recognition is focused on classifying data into classes based on specific attributes [19]. Machine learning and deep learning are techniques used to define algorithms, and each uses data to learn [18-20]. Machine learning uses supervised and unsupervised methods to discover and model the patterns and relationships in data, allowing it to make predictions. Deep learning uses machine-learning approaches to automatically learn and extract features from complex unverified data without human involvement [18, 20]. Artificial neural networks, conceptually inspired by how the human brain works using biological neurons, are models trained on past data to make predictions [19]. The degree of human intervention in the decision-making process varies according to the type and purpose of the integration [18]. Technologies such as big data, predictive analytics, business intelligence, advanced analytics, and some digitization projects provide the foundation for these solutions. The technical processes for building algorithms require high-performance computing systems and architectures [18].

C. Algorithmic decision-making

Algorithmic decision-making can be viewed as having three stages: development, usage, and consequence [8, 18]. The development stage produces an algorithmic system in three steps. The source data are collected from multiple sources; the data are made fit for purpose, including augmenting it with tags, identifiers, or metadata; and stored in data repositories. For the second step, subsets of source data are transformed into data for training the models (referred to as training data). The models and algorithms are developed through the extensive use of data and analytical methods. This activity is training the model. Here high-performance computing is needed to support the computational load and data volumes. The algorithms are validated. A user interface is developed for producing autonomous decisions or providing input for human decision-making. This step may include other technical aspects, such as system deployment; these topics are relevant but not the main focus of this study. In the usage stage, the algorithms are used by inputting parameters or data to invoke them; the algorithms output the decisions. The algorithm or systems may be standalone systems, integrated into other systems, robots, automobiles, etc., or exists in a digital technology platform such as a social media platform. In the consequences stage, the decision is finalized, and the consequences are realized on people, organizations, and groups.

D. Morality and ethics in AI

Jones [21] defines a moral issue as one where a person's actions, when freely performed, has consequences (harms or benefits) on others. The moral issue must involve a choice on the part of the actor or the decision-maker. He summarizes that many decisions have a moral component as they affect others. A moral agent is a person that makes the decision even when the decision-maker may not recognize a moral issue is at stake. An ethical decision is both legally and morally acceptable to the larger community; an unethical decision violates either the legal or the moral acceptability. Much of the research reviewed, treat the terms moral and ethical as equivalent and use them interchangeable depending on the context.

The thesis from [22] on morality is that the concepts of right and wrong should be discarded and replaced with a definition of morality in terms of "intrinsically unjust" versus "unjust given the circumstances." He argues that the boundary between the two concepts is "according to what's reasonable." Anscombe [22] further theorizes that determining the expected consequences plays a part in determining what is just. These arguments place the responsibility for morality on the decision-maker. However, they do not answer who is accountable when the decisions are delegated from humans to systems.

Manders-Huits [7] argues that the notion of consequences and level of autonomy of action are preconditions or considerations for moral responsibility. First, the notion of consequences in information technology (IT) places the burden of responsibility for moral decisions on the shoulders of the designers of complex IT systems. However, the definition of the designers is unclear—technicians, finance providers, instructors—as well as how the designer's responsibility relates to the responsibility the end users have for final decision-making. Martin [8] also places the responsibility for moral decision-making with the system developer and their companies. Second, the abundance of information that individuals have and understand enhances their possibility of action autonomy. The actions or decisions integrated into IT applications are limited based on "implying an adequate understanding of all relevant propositions or statements that correctly describe the nature of the action and the foreseeable consequences of the action" [7]. It is not likely that modelers can predict all potential uses of their models [23]. Consequently, [24] argues that machines are artificial agents that should not be held to a higher moral standard than humans and define five meta-moral qualities that machines should possess to be considered proper moral agents (robustness, consistency, universality, and simplicity).

A significant amount of research has focused on defining values, principles, frameworks, and guidelines for ethical AI development and deployment [13, 25]. However, [13] determined that principles alone have a limited impact on AI design and governance. Conducting an analysis of 21 AI ethic guidelines, [26] similarly found that AI guidelines are ineffective and do not change the behavior of professionals from the technology community. One challenge is the difficulty in translating concepts, theories, and values into practice. Specifically, the translation process is likely to "encounter incommensurable moral norms and frameworks which present true moral dilemmas that principles cannot resolve" [13]. Furthermore, there are no proven methods to translate the principles into practice. Mittelstadt [13] warns that the solution to AI ethics should not be oversimplified to addressing only the AI technical design or expertise.

Jobin, et al. [25] conducted a content analysis of 84 AI ethical guidelines and identified five ethical principles that converged globally (transparency, justice, fairness, nonmaleficence, and privacy). Building on the research from [25], [27] provides a detailed explanation of the normative implication of AI ethics guidelines for developers and organizational users. The paper provides a deep dive into the details. It specifies AI ethical principles and what users and developers ought to do to realize their moral responsibilities. However, the study explicitly excludes other stakeholders. Furthermore, in providing AI ethics research, [28] identified that AI ethics interests change over time.

III. Research Methodology

This section describes the research methodology, including the theoretical model.

A. Theoretical Framework

To answer the research question, this research seeks the deliverables, acts or situations necessary to avoid harm or ensure benefits of an algorithm developed in projects. Thus, the project success model from [9] is relevant for identifying the success factors. It attempts to forecast project success beyond just the initial project outputs. It recognizes multiple stakeholders interested in the project output, outcomes, impacts, and that stakeholder interest change over time.

The model from [9] was chosen for four key reasons. First, the model focuses on projects and projects are bound by time, team, tasks, and activity. These boundaries limit environmental considerations. This is relevant as personal experience, organizational norms, industry norms, and cultural norms affect stakeholders' perceived alternatives, consequences, and importance. Second, decisions made during the project will have an impact many months or years in the future. However, the project participants may not be aware of the magnitude of the consequences of their decisions in terms of harms or benefits on their victims or beneficiaries at the time of the decision. Thus, it is important to consider the multiple time dimension available in the model. Third, stakeholders influence the project's planning and outputs and are impacted by the project results. Thus, the multiple stakeholder perspectives are useful for considering the influence of the decision-making and the impact of the decisions on the stakeholders. Finally, the model outlines the multiple types of success indicators that should be considered in the investigation.

The algorithmic development, usages, and consequence stages and AI components were aligned with the timescales with the model from [9]. Algorithm development aligns with the project output, algorithm usage with the outcome, and decision-making consequences with the impact. Table I identifies the alignment of AI components to the time scales.

B. Systematic Review Procedure

A systematic review of the literature was used to explore the research question. "A systematic review is a review of a clearly formulated question that uses systematic and explicit methods to identify, select, and critically appraise relevant research, and to collect and analyze data from the studies that are included in the review" [29]. The purpose of the systematic review was to synthesize existing knowledge in a structured and rigorous manner. The procedure included an 1) identification of bibliographic databases from which to collect the literature, 2) definition of the search process including the keyword and the search string, 3) definition of inclusions and exclusion criteria, 4) removing duplicates and screening the articles, 5) extracting data based on a full-text review of the articles, and 6) synthesizing the data using a coherent coding method. Details are described in the following sections, and Fig. 1 includes a flow of information through the systematic review. The process was conducted by a single researcher.

I. Bibliographic databases

The first literature search was in October 2020 for peer-reviewed articles in the ProQuest, Emerald, ScienceDirect, and IEEE Xplore bibliographic databases. The focal keywords were "algorithm" and "stakeholder." This search revealed key themes in how success was viewed in algorithmic projects. Keywords such as ethics, fairness, accountability, transparency, and explainability were frequently referenced in the articles. The analysis identified the "ACM Conference on Fairness, Accountability, and Transparency (ACM FAccT)" as an important source for cross-disciplinary research. Thus, bibliographic searches were undertaken in March and July 2021, adding the ACM Digital Library to the previous bibliographic databases.

II. Search string

The ultimate search was performed, placing emphasis on "accountability" instead of "stakeholder." Stakeholder in combination with algorithm was not a frequent keyword, and accountability focuses on the relationship between project actors and those to whom the actors should be accountable [30]. Other frequent keywords were also included in the search string to make the results meaningful. Since not all databases allowed wild cards, variations of the search string were used, and adjustments were made in the syntax for each search engine. The wildcard version of the search string is as follows.

TABLE I.
PROJECT TIMESCALES AND AI COMPONENTS

Time scales	**AI component**
Output	• Source Data, Data Collection and Storage • Training Data • Model and Algorithm Development • Model and Algorithm Validation • User Interface • System Architecture & Configuration
Outcome	• Input Interface • Model and Algorithm Usage
Impact	• Decisions

All=accountabl* AND

Title = ("machine learning" OR "artificial intelligence" OR AI OR "big data" OR algorithm*) AND

Title = (fair* OR ethic* OR moral* OR success OR transparency OR explainabl*)

III. Inclusions and exclusion criteria

Articles were retained in the search result for peer-reviewed journal articles or conference papers and English language; book reviews were excluded. There were no filters for the dates. Duplicate entries and entries with no document were removed. Next, literature was excluded or retained in an iterative process based on first a review of the title, second a review of the abstract, and finally a review of the full article text.

First, the title of the articles was reviewed and articles were retained that were about the process or considerations for the development, use, or outcomes of algorithms. Articles were excluded that were about the structure or content of an algorithm, a specific use case, or wrongly identified articles, e.g., magazine articles, panel descriptions. Next, the abstracts were reviewed to determine if the article could yield information on the success factors. Finally, the full text of the included articles was reviewed and coded to answer the research question. New articles identified during the analysis process were manually added. In total, the full-text of 85 articles were included for analysis. The majority of the included articles (79%) were published since 2019 and many (36%) were conference papers. Table II shows the article distribution by database and Fig. 1 shows the preferred reporting items for systematic reviews and meta-analyses (PRISMA) process flow.

IV. Data analysis

Each of the 85 articles was reviewed in detail for coding the success factors. The coding was conducted in Nvivo 12 (Windows) software. We extracted terms and explanations to determine what was known about how different stakeholders viewed success; we used the literature to clarify the definitions, provide examples, determine the main elements of success, and develop context. We compiled a resulting list of success factors that had to be deliverable, acts or situations that contributed to a positive outcome with algorithm decision-making projects. The success factors were qualitatively grouped based on their common characteristics or responsibility patterns. The results are summarized in the Research Findings section.

C. Validity and Reliability

This approach of defining elements is an acceptable method for placing boundaries around the meaning of a term [29]. First, internal consistency was provided by using a theoretical model to conduct the literature search and produce the guiding questions. Second, external validity was ensured by using literature as a primary source and a validation source. The success factors were mapped at a detailed level to their original sources in the literature. The results were cross-validated with prior AI ethic literature reviews from [27] to ensure completeness. The checklist and phase flow from the PRISMA Statement were used to guide the study and report the results [29].

IV. Research Findings

The literature review identified 71 success factors that were qualitatively consolidated in an iterative process into three broad categories and 14 groups. The results describe the practical requirements for success with AI development and usage based on the moral issues and ethical principles found in the literature. From an AI development perspective, the factors align with each of the AI components and translate principles into design and development requirements. For operations, the factors are the procedures for the usages of the algorithms and for addressing the concerns and expectations of the stakeholders. From a project point of view, the factors are the management concerns of end users, project sponsors, and project managers. Thus, the principles of trustworthiness, transparency, explainability, accountability, sustainability, etc., are distributed throughout the individual success factors.

First, features, capabilities or content of the deliverables were categorized as belonging to product qualities. Expectations, processes, or procedures related to the content and usage of the deliverables are placed in a procedure category. The third category of success factors relates to the management process, benefits, or protections expected. These categories align with the conduct groups in the accountability model referenced in [30].

Table II.
Article distribution by Database

Database	Search Results	Duplicate	Screened by Title	Abstract	Eligible
ACM	172	10	162	139	23
Emerald	8	4	4	2	1
IEEE Xplore	117	8	109	20	10
ProQuest	118	14	104	96	31
Science Direct	74	2	72	16	6
Manual	20		20	9	14
	509	38	471	282	85

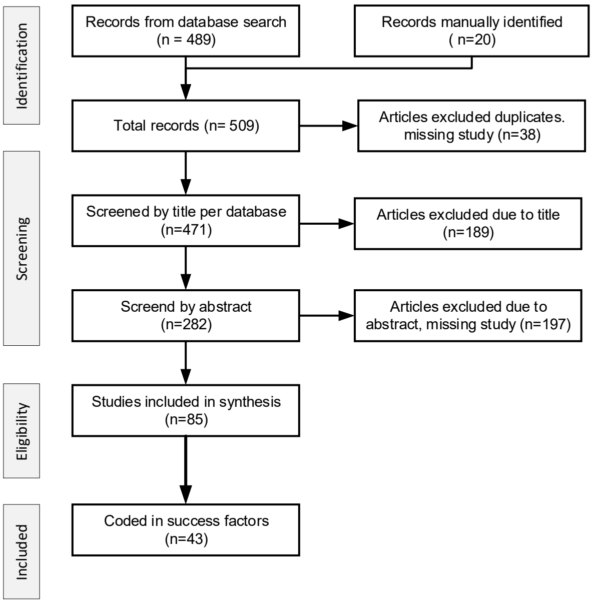

Identification
Screening
Eligibility
Included
Records from database search (n = 489)
Records manually identified (n=20)
Total records (n= 509)
Articles excluded duplicates. missing study (n=38)
Screened by title per database (n=471)
Articles excluded due to title (n=189)
Screend by abstract (n=282)
Articles excluded due to abstract, missing study (n=197)
Studies included in synthesis (n=85)
Coded in success factors (n=43)

Fig 1. PRISMA process flow

The characteristics of the factors or their impact or influence by project actors or stakeholders influenced the categorization. Table III identifies the success factors based on their categories and groups. This section describes each success factor by category and group; the factors are italicized in the text.

A. Product Qualities

I. Source Data Qualities

Data accessibility refers to access and use of data in the algorithm creation process. Several regulations and laws constrain how data may be accessed, processed, and used in analytical processes. Thus, a legal agreement to use the data and confidentiality of personal data should be preserved [1, 3, 31-35]. *Data transparency* refers to revealing the source of the data collected, including the context or purpose of the data collection, the application, the sensors (or users that collected the data), and the location in which the data are stored [23, 33, 36-40]. The reviewability framework [41] recommends maintaining *data collection records* of data and their lifecycle. The recommended content includes providing details on purpose, creators, funders, composition, content, collection process, usage, distribution, limitations, maintenance, and data protection and privacy concerns [2, 33, 34, 36, 38, 41]. Datasheet by [36] provides detailed guidance on document content.

II. Training Data Qualities

In interacting with and processing data, individuals are entitled to physical and psychological safety, i.e., *interaction safety* [3, 23, 27, 33, 35, 42, 43].

TABLE III.
SUCCESS GROUPS AND SUCCESS FACTORS

Category	Success Groups	Success Factors	References
Product Qualities	Source Data Qualities	Data accessibility, Data transparency, Data collection records	[1-3, 23, 31-41]
	Training Data Qualities	Data quality and relevance, Interaction safety, Equitable representation, Model training records	[3, 23, 27, 33, 35, 42-46]
	Models & Algorithms Qualities	Algorithm transparency, Consistency, Accuracy, Interpretability, Auditability, Model validation, Algorithm renewal, Model validation records	[1-3, 23, 27, 31, 33, 37, 38, 45-52]
	User Interface Qualities	Human intervention, Equitable accessibility, Front-end transparency	[1, 23, 27, 31, 32, 34, 37, 39, 53, 54]
	System Configuration	System and architecture quality, Security safeguards, Technical logging, Technical deployment records	[2, 3, 31, 41, 49, 55]
	Data Privacy & Confidentiality	Informed consent, Personal data controls, Confidentiality, Privacy safeguards, Data anonymization, Data encryption, Data retention policy	[1, 3, 23, 27, 31-33, 37]
Procedures	Decision Quality	Awareness, Access and redress, Decision accountability, Equitable treatment, Privacy and confidentiality, Civil rights and liberty protections	[1-3, 23, 27, 32, 34, 37, 40, 42, 44, 49, 56]
	System Transparency & Understandability	User-centric communication, Interpretable models, Choices, Specialized skills and knowledge, Interaction safety, Problem reporting, Usage records	[2, 23, 27, 31, 33-35, 37, 40, 41, 49, 57-59]
	Usage Controls	Compliant process, Quality controls, Monitoring, Consequence records, Process deployment records	[23, 34, 39, 41, 51, 59]
	Investigation	Algorithm auditing, Audit finding records, Audit response records, Algorithm impact assessments, Certification	[1-3, 23, 30, 41, 47, 51, 60]
Management	Governance	Scope definition document, Responsibility assignment matrix, Diverse working environment, Ethics policies, Recordkeeping, Risk assessment records, Disclosure records, Procurement records	[10, 27, 34, 41, 61-63]
	Financial Benefits	Intellectual property rights, Profits, License or service fees, Investment funds	[23, 40, 42]
	Financial Protections	Intellectual property protection, Environmental impacts, Energy costs, Cost efficiency, Project efficiency	[9, 14, 27, 42, 46, 47]
	Legal Protections	Limiting liability, Legal safeguards, Regulatory and legal compliance	[2, 3, 23, 27, 31, 33, 40, 41, 51, 60]

Equitable representation applies to data and people. For data, it means having enough data to represent the whole population for whom the algorithm is being developed while also considering the needs of minority groups such as handicapped people, minors (under 13 years old), and ethnic minorities. For people, it means, for example, including representatives from minority groups or their advocates in the project governance structures or teams that design and develop algorithms [23, 44-46]. *Model training records* should document the training work flow, model approaches, predictors, variables, and other factors; datasheets by [36] and model cards by [48] provide a framework for the documentation.

III. Model & Algorithm Qualities

Algorithm transparency refers to using straightforward language to provide clear, easily accessible descriptive information (including trade secrets) about the algorithms and data and explanations for why specific recommendations or decisions are relevant. The need for end users to understand and explain the decisions produced by the algorithms determines the algorithm, data, and user interface transparency requirements [1, 23, 31, 33, 37, 47, 48]. Model qualities include consistency, accuracy, interpretability, and suitability; there are no legal standards for acceptable error rates or ethical designs. *Consistency* means receiving the same results given the same inputs; nondeterministic effects can occur based on architectures with opaque encodings or imperfect computing environments [3]. *Accuracy* is how effective the model provides the desired output with the fewest mistakes (e.g., false positives, error rates) [3, 23, 37, 45, 46]. *Interpretability* refers to the degree to which the model is designed to provide reliable and easy-to-understand explanations of its prediction [27, 37, 49]. *Auditability* refers to how the algorithm is transparent to or obfuscated from an external view to allow other parties to monitor or critique it [2, 38].

Model validation is the execution of mechanisms to measure or validate the models for adherence to defined principles and standards, effectiveness, performance in typical and adverse situations, and sensitivity. The validation should include bias testing, i.e., an explicit attempt to identify unfair bias, avoid individual and societal bias, and reverse any biases detected. Models can be biased based on a lack of representations in the training data or how the model makes decisions, e.g., the selected input variables. The model outcomes should be traceable back to input characteristics [2, 23, 50-52]. Model values or choices become obsolete. They need to be reviewed or refreshed so an *algorithm renewal* process should be established [23, 30]. The reviewability framework suggests maintaining *model validation records* that contain details on and how the model was validated, including dates, version, intended use, factors, metrics, evaluation data, training data, quantitative analyses, ethical considerations, caveats and recommendations, or any other restrictions [41, 48]. Model cards by [48] provide detailed guidance on the content.

IV. User Interface Qualities

Expertise is embodied in a model in a generalized form that may not be applicable in individual situations. Thus, *human intervention* is the ability to override default decisions [1, 34, 37]. *Equitable accessibility* ensures usability for all potential users, including people with disabilities [23, 27, 53]. *Front-end transparency* designs should meet transparency requirements and not unduly influence, manipulate, confuse, or trick users [31, 32, 39, 54]. Furthermore, dynamic settings or parameters should consider context to avoid individual and societal biases such as those created by socio-demographic variables [34]. App-Synopsis by [54] provides detailed guidance on the content.

V. System Configuration

The *system and architecture quality* may impact the algorithm's outcomes, introduce bias, or result in indeterminate behavior. Default choices (e.g., where thresholds are set and the defaults to be specified) may introduce bias in the decision-making. Specifically, the selected defaults may be based on the personal values of the developer. Decisions on methods and the parallelism of processes may cause system behavior that does not always produce the same results when given the same inputs. Obfuscated encodings may make it difficult to process the results or audit the system. The degree of automation may limit the user's choices [3, 49]. *Security safeguards* are implementing technology, processes, and people to resist accidental, unlawful, or malicious actions that compromise the availability, authenticity, integrity, and confidentiality of data [2, 31, 55].

The reviewability framework suggests the systems should provide a *technical logging* process including mechanisms to capture the details of inputs, outputs, and data processing/computation. The framework also recommends records relevant to the *technical deployment records* and operations, including installation procedures, hardware, software, network, storage provisions or architectural plans, system integration, security plans, logging mechanisms, technical audit procedures, technical support processes, maintenance procedures [41].

B. Procedures

I. Data Protection, Privacy and Confidentiality

Informed consent is the data subject's right to be informed on the collection, use, and repurposing of their personal data [3, 23, 27, 31, 37]. The legal and regulatory rules covering consent vary by region and usage purposes. *Personal data control* means giving people control of their personal data [1, 32, 37]. *Confidentiality* concerns protecting and keeping confidential data and proprietary information. *Privacy safeguards* include processes, strategies, guidelines, and measures to protect and safeguard data privacy, along with remedies for privacy breaches. For example, a privacy measure could be data encryption or data anonymization [1, 3, 23, 32, 33, 37]. *Data anonymization* involves applying rules and processes that randomize data so an individual is not personally identifiable and cannot be re-identified through combining data sources. In general, data protection principles do not apply to anonymous information [23, 32, 33]. *Data encryption* is an engineering approach to secure data with electronic keys. *Data retention policy* specifies the time and obligations for keeping data [31].

II. Decision Quality

Awareness is educating the public about the existence and the degree of automation, the underlying mechanisms, and the consequences [2, 37]. *Access and redress* are a way to investigate and correct erroneous decisions. It includes the ability to contest automated decisions, including expressing a point-of-view or requesting human intervention in the decision [1, 2, 27, 37, 40, 56]. *Decision accountability* is knowing who is accountable for the actions when decisions are taken by the automated systems in which the algorithms are embedded [2, 27, 56]. *Equitable treatment* means eliminating discrimination and differential treatment, whereby similarly situated people are given similar treatment. In this context, discrimination does not only equate to prejudice based on race. It is based on forming groups using 'statistical discrimination'; it further refers to anti-discrimination and human rights protections [1, 2, 27, 34, 42, 44]. *Privacy and confidentiality* are the activities for protecting and keeping confidential information of an identified or identifiable natural person [3, 23, 27, 34, 40, 42, 49]. In this context, *civil rights and liberties protection* are securing and providing the fundamental rights and freedoms of natural persons, including the right to data protection and privacy and to have opinions and decisions made independently of an automated system [27, 32].

III. System Transparency & Understandability

User-centric communication considers the explainability of the algorithm to the intended audience. It transmits essential, understandable information rather than legalistic terms and conditions. Explanations are communicated in layman's terms, even for complex algorithms [2, 27, 31, 34, 37, 40, 57]. *Interpretable models* refer to having a model design that is reliable, understandable, and possible for expert users to explain the predictions [27, 37]. *Choices* allow users to decide what to do with model results or, in other words, provides a degree of human control [23, 27, 37, 49, 58].

Expertise is embodied in a generalized form that may not be applicable in individual situations, so *specialized skills and knowledge* may be required to choose between alternatives. Consequently, professional expertise, staff training and supervision, and on-the-job coaching may be necessary to ensure appropriate use and decision quality [49, 59].

Interaction safety refers to ensuring physical and psychological safety for the people interacting with the AI systems [35]. *Problem reporting* is a mechanism that allows users to discuss and report concerns such as bugs or algorithmic biases [47]. The reviewability framework recommends retaining *usage records* of model inputs and outputs of parameters, operational records at the technical (systems log) level, usage instructions [33, 41].

IV. Usage Controls

The *complaint process* means having mechanisms in place for identifying, investigating, and resolving improper activity or receiving and mediating complaints [39]. *Quality controls* detect improper usage or under-performance. Improper usage occurs when the system is used in a situation for which it was not originally intended [23, 34]. *Monitoring* is a continual process of surveying the system's performance, environment, and staff for problem identification and learning [59]. System monitoring is to verify how the system behaves in unexpected situations and environments. The staff monitoring identifies absent or inadequate content areas, identifies systematic errors, anticipates and prevents bias, and identifies learning opportunities.

The reviewability framework recommends retaining consequence and process deployment records. *Consequence records* document the quality assurance processes for a decision and log any actions taken to affect the decision, including failures or near misses [41, 51]. Logging and recording decision-making information are appropriate means of providing traceability. *Process deployment records* document relevant operational and business processes, including workflows, operating procedures, manuals, staff training and procedures, decision matrices, operational support, maintenance processes and records [41].

V. Investigations

Algorithm auditing is seen as a method for understanding how algorithms work. Testing algorithms based on issues that should not arise and making inferences from the algorithms' data is a technique for auditing complex algorithms [1-3, 41, 47, 51]. Audit records include audit finding records and audit response records. *Audit finding records* document the audit, the basis or other reasons it was undertaken, how it is conducted, who conducted it, any findings [41]. *Audit response records* document remediations and subsequent actions or remedial responses based on audit findings [2, 41].

Algorithmic impact assessments investigate aspects of the system to render visible impacts of the systems and propose steps to address any deficiencies or harms [30, 60]. *Certification* identifies that people or institutions comply with regulations and safeguards and publicize institutions with breaches; it offers independent oversight by an external organization [23, 51].

C. Management

I. Governance

The *scope definition document*, or problem statement, defines the aims and rationale for the algorithmic system [10, 41]. The requirement for the system, the moral issues, and all aspects of the project are impacted by the context (country, industry sector, functional topic, and use case) of the algorithm. Trust is context-dependent since things can work in one context but not another; thus, the scope should act as a contract that makes explicit the algorithm's goal and the behavior that can be anticipated [61]. Furthermore, a clearly defined scope protects against spurious claims and misapplication or misuse of the system. Next, ethical principles argue AI systems should be developed to do good or benefit someone or the society as a whole (beneficence); they should avoid doing harm to others (non-maleficence) [27, 34]. Finally, rules should be established on managing conflict of interest situations within the team or when the values of the system conflict with the interests or values of the users [62, 63].

A *responsibility assignment matrix* defines roles and responsibilities within a project organization. It distinguishes between persons or organizations with responsibility and accountability [64]; accountability ensures a task is satisfactorily done, and responsibility accepts an obligation to perform a task satisfactorily, with transparency in reporting on outcomes, corrective actions, or interactive controls [64, 65]. Both responsibility and accountability assume a degree of subject matter understanding and knowledge [27]. The project organization should promote a *diverse working environment*, including involving various stakeholders and people from differing backgrounds and disciplines and promoting the exchange and cooperation across regions and between organizations [27, 43].

Ethics policies should include guidelines and rules for implementing, verifying, and remedying ethical principles; the guides should be shareable externally with the public or public authorities [2, 27, 33]. The practical aspects of ethical principles for fairness, trustworthiness, transparency, explainability, accountability, and sustainability are distributed throughout the individual success factors discussed in this study.

Systematic *recordkeeping* is the mechanism for retaining logs and other documents of contextual information about the process, decisions, and decision-making from the project inception through the system operations [10, 27, 33, 41, 49]; the various types of records are recorded as individual success factors. The *risk assessment records* identify the potential implications and risks of the system such as legality and compliance, discrimination and equality, impacts on basic rights, ethical issues, sustainability concerns [10, 41]; *disclosure records* are logs that are themselves about disclosures or the processes for disclosure, what was actually released, how information was compiled, how it was delivered, in what format, to whom, and when [31, 33, 41]; and *procurement records* are contractual arrangements, tender documents, design specifications, quality assurance measures, and other documents that detail the suppliers and relevant due-diligence [41].

II. Financial Benefits

Intellectual property rights consist of the ownership of the design of the models, including the indicators. Innovation levels have to be balanced with risks of liabilities and litigation for novel concepts [23]. *Profits* include increased revenues from the sale or licensing models that produce revenue through *license or service fees* [23, 42] or reductions in costs from making faster, less expensive, or better decisions [40]. Furthermore, proven successful models, concepts, algorithms, or business models can attract *investment funds* [23].

III. Financial Protections

Intellectual property protection is achieved by partly or entirely hiding the algorithm's design choices. Data and algorithm transparency and auditing requirements should be considered in deciding what to reveal [47]. Model development has environmental impacts and energy costs. The *environmental impacts* occur as the big training models may be energy-intense using as much computing energy as a trans-American flight in carbon emissions [27, 46]. The *energy costs* from computing power and electricity consumption (for on-premise or cloud-based services) are relevant for training models [27, 46]; for an incremental increase inaccuracy, the cost of training a single model may be extreme (e.g., 0.1 increase in accuracy for 150,000 USD) [46]. *Cost efficiency* occurs acquiring and using information is less than the costs involved if the data were absent [42]. *Project efficiency* evaluates the project management's success in meeting stakeholder requirements for quality, schedule, and budget [9, 14].

IV. Legal Protections

The *legal safeguards* include protection against legal claims or regulatory issues that arise from algorithmic decisions [2, 31]. *Limiting liability* or risk of litigation for users and balancing risks from adaptations and customizations with fear of penalties or liability in situations of malfunction, error or harms [23, 27]. *Regulatory and legal compliance* involves meeting the legal and regulatory obligations for collecting, storing, using, processing, profiling, and releasing data or complying with other laws, regulations, or ordinances [3, 33, 40, 41, 51, 60].

V.Discussion

This study framed the question of project success from the perspective of moral decision-making with algorithms. People impacted by algorithm decisions want fairness, meaning moral or "just" treatment from algorithmic decision-making. However, fairness or the perception of fairness has several subjective components that are out of the scope of any development project, including pre-established attitudes and emotional reactions to algorithmic outcomes [1, 44]. Moreover, [66] empirically found that end users understand, perceive, and process algorithm fairness, accountability, and transparency differently. Furthermore, the interaction between trust and algorithmic features influence user satisfaction.

Nevertheless, the research revealed that the project team's actions influence who is the judge of what is reasonable when the decision is made [7]. Thus, the project team has some responsibility for the moral decisions produced by the algorithmic systems. The limits and bias in decisions produced, the end user's ability to manipulate the system or override the decisions, and the information the end users have to understand and enhance their decision autonomy mediate the project team's accountability. The project organization can take on some responsibility by considering moral decision-making in the project scope. The success factors for the product qualities, procedures, and management are discussed in the following sections.

A. Product Qualities

The product quality success factors must be considered from many external stakeholders' perspectives, including individuals, society, end users, user organizations, technology platforms, etc. Thus, each development aspect needs to consider the technical product qualities, usability features, information requirements, and legal and regulatory requirements. In this regard, several conflicting success factors have to be balanced. For example:

- The end users may want a high degree of flexibility for human intervention, including making alternative choices. Similarly, the person impacted by the decision outcome will want to have erroneous (or biased) decisions reviewed and corrected. Conversely, the user's organization would want to limit legal liabilities, which speaks for fewer choices. The more open the system, the harder it is to differentiate between a system error and user error and assign accountability.
- The need for the end users to understand and explain the decisions produced by the algorithms suggests a high degree of transparency for the algorithm, data, and front-end user interface. Conversely, the need to preserve intellectual property rights is a factor for a lesser degree of transparency.
- Unbiased models can produce high error rates (or be inaccurate), and biased models can be accurate. Thus, there is a tradeoff between utility and fairness due to bias or inaccuracies.
- There is a tradeoff between the degree of automation and human autonomy. Too much automation can give the perception (or reality) that people are under constant surveillance or that the system knows too much and is what [40] calls creepy. Meanwhile, the system can offer flexibility, accuracy, or benefits not available through human autonomy.
- Developing large-scale language models produces carbon emissions and has a financial cost. However, the assumption (which is challenged) is that large models

increase accuracy. Thus, there is a tradeoff between accuracy, environmental impacts, and financial costs.

The success factors in the data protection, privacy and confidentiality group relate to the product qualities and usage procedures. Specific product capabilities may be needed to realize certain processes. For example, personal data controls require some degree of system traceability and extracts for personal data. Also, it must be clear when data stops being personal within the system and becomes generic or anonymized. Another overlap between product qualities and procedures relates to the methods and practices used to implement privacy controls or data anonymization.

B. Procedures

The procedures for using and investigating algorithmic systems depend on the many product qualities and procedures enabled by the system or implemented by the end users and their organization. The user's organization and the platform providers must follow regulations and laws relevant to the industry, data processing, and data profiling. Furthermore, as of April 2021 in the European Union, the artificial intelligence regulation act requirements should be considered [67]. Thus, success factors are robust operational rules, policies, contracts; quality controls; and privacy and security safeguards.

C. Management

There are several success factors from a business and governance perspective for delivering the product, intellectual property rights and protections, limiting liability, ensuring legal safeguards and regulatory compliance. Similar to the product qualities, there are multiple conflicting success factors. For example:

- The tradeoff between accuracy, environmental costs, and financial costs is already discussed.
- The need for financial profits from algorithm systems and the need to benefit society (beneficence) may result in conflicting objectives.
- The project efficiency concerning quality, time, and budget and the regulatory and legal compliance.
- The need for algorithm, data, and front-end user interface transparency and producing intellectual property rights and protections.
- The need for legal safeguards, comparing the need for system flexibility to allow for choices at the point of decision versus restricting human intervention.

VI. Conclusions

The importance of algorithms in society and individuals' lives is becoming increasingly apparent. Therefore, the success factors for AI projects are important and are dramatically more expansive than those for a typical information systems project. This research identified 71 AI project success factors in 14 groups related to moral decision-making with AI projects. The research summarizes the concerns for fair, moral algorithm development and usage in decision-making. It reveals the project manager and the project team need to consider many factors when defining the project scope and executing it. An AI development project has a narrower scope and fewer short-term implications than an algorithmic development project considering moral decision-making. This paper argues that the people that develop and operate AI systems are moral agents. Those people should build AI systems and procedures to avoid harm and ensure benefits. Hence, as artificial agents, the systems should abide by the moral laws of society.

Projects are constrained by time and budget, limiting the availability of people and other resources. Nevertheless, the importance of the algorithms that result from AI projects can be significant. Thus, it is necessary and relevant that a broad view of success factors be considered in planning and executing these projects. The findings from this study provide some guidelines on the success factors that may only be used indirectly or overtime to judge the project's success.

A. Practical Implications

Projects, and especially AI projects, are context-sensitive. The factors presented are generic; it would be important to adjust and validate in specific contexts. For example, developing an algorithm for a healthcare situation would have different considerations than an algorithm for a marketing situation.

The success factors provide insights into the activities and deliverables that should be considered part of the planning to ensure fair, ethical decisions. First, the project manager and sponsor should alter project scopes to consider moral decision-making with algorithms. This will dramatically affect the team compositions and the deliverables produced as part of the project. The benefits to society and the environment could be highlighted and potentially measured.

Next, project managers and sponsors may be limited in influencing future usage and operational processes. Nevertheless, they should try to exert this influence on the ethical practices of system users and user organizations. Furthermore, they should consider the success factors described herein to recognize moral issues that require decisions during the development process to mitigate project risks. Finally, as an agent of the sponsoring organization with a reputation to manage and business objectives to reach, the project manager should consider these success factors to ensure adherence to ethical, privacy, and security norms and deliver business benefits.

B. Theoretical Implications

The research expands the existing project management literature on project success factors specific to the AI domain. This contribution is consistent with the direction identified by [14], "...we argue that one should turn to context-specific and even symbolic and rhetoric project success and CSFs [critical success factors]."

C. Limitations and Future Research

This research was based on the latest available literature, but at a single point in time. AI is a fast-moving topic judging from the number of recent articles. Thus, other methods such as a Delphi study with field experts could extend and update the study and validate the findings. Since the analysis was conducted by a single researcher, the results may be biased by the researcher's perspective.

As an opportunity for additional research, the success factors could be used to investigate project accountability or stakeholder management. It could be expanded to identify measurable success criteria for some success factors. AI literature regarding ways to measure bias, inequality, and accuracy should be left to the specialists; however, it would be interesting to understand how to evaluate the tradeoffs needed during the projects and still meet all stakeholder requirements retaining an honest approach.

REFERENCES

[1] N. Helberger, T. Araujo, and C. H. de Vreese, "Who is the fairest of them all? Public attitudes and expectations regarding automated decision-making," *Computer Law & Security Review,* vol. 39, pp. 1–16, Nov 2020.

[2] S. Garfinkel, J. Matthews, S. S. Shapiro, and J. M. Smith, "Toward algorithmic transparency and accountability," *Commununicaions of the ACM,* vol. 60, p. 5, 2017.

[3] J. A. Sherer, "When Is a Chair Not a Chair?: Big Data Algorithms, Disparate Impact, and Considerations of Modular Programming," *Computer and Internet Lawyer,* vol. 34, pp. 6–10, Aug 2017.

[4] S. Baruffaldi, B. v. Beuzekom, H. Dernis, D. Harhoff, N. Rao, D. Rosenfeld, *et al.*, "Identifying and measuring developments in artificial intelligence," 2020.

[5] A. J. Shenhar, D. Dvir, O. Levy, and A. C. Maltz, "Project success: a multidimensional strategic concept," *Long range planning,* vol. 34, pp. 699-725, 2001.

[6] S. J. Bennett, "Investigating the Role of Moral Decision-Making in Emerging Artificial Intelligence Technologies," presented at the Conference Companion Publication of the 2019 on Computer Supported Cooperative Work and Social Computing, Austin, TX, USA, 2019.

[7] N. Manders-Huits, "Moral responsibility and IT for human enhancement," presented at the Proceedings of the 2006 ACM symposium on Applied computing, Dijon, France, 2006.

[8] K. Martin, "Ethical Implications and Accountability of Algorithms:," *Journal of Business Ethics,* vol. 160, pp. 835–850, Dec 2019.

[9] R. J. Turner and R. Zolin, "Forecasting Success on Large Projects: Developing Reliable Scales to Predict Multiple Perspectives by Multiple Stakeholders Over Multiple Time Frames," *Project Management Journal,* vol. 43, pp. 87––99, 2012.

[10] O. Zwikael and J. R. Meredith, "Who's who in the project zoo? The ten core project roles," *International Journal of Operations & Production Management,* vol. 38, pp. 474–492, 2018.

[11] K. Davis, "An empirical investigation into different stakeholder groups perception of project success," *International Journal of Project Management,* vol. 35, pp. 604–617, May 2017.

[12] R. K. Mitchell, B. R. Agle, and D. J. Wood, "Toward a Theory of Stakeholder Identification and Salience: Defining the Principle of who and What Really Counts," *Academy of Management Review,* vol. 22, pp. 853-886, Oct 1997.

[13] B. Mittelstadt, "Principles alone cannot guarantee ethical AI," *Nature Machine Intelligence,* vol. 1, pp. 501-507, 2019.

[14] L. A. Ika, "Project success as a topic in project management journals," *Project Management Journal,* vol. 40, pp. 6--19, 2009.

[15] C. Weninger, "Project Initiation and Sustainability Principles: What Global Project Management Standards Can Learn from Development Projects when Analyzing Investment," presented at the Paper presented at PMI® Research and Education Conference, Limerick, Munster, Ireland, 2012.

[16] J. K. Pinto and D. P. Slevin, "Critical Success Factors Across the Project Life Cycle," *Project Management Journal,* vol. 19, p. 67, 1988.

[17] W. Belassi and O. I. Tukel, "A new framework for determining critical success/failure factors in projects," *International Journal of Project Management,* vol. 14, pp. 141-151, 1996.

[18] G. J. Miller, "A conceptual framework for interdisciplinary decision support project success," in *2019 IEEE Technology & Engineering Management Conference (TEMSCON)*, 2019, pp. 1-8.

[19] J. Aggarwal and S. Kumar, "A Survey on Artificial Intelligence," *International Journal of Research in Engineering, Science and Management* vol. 1, Dec 2018.

[20] R. Iqbal, F. Doctor, B. More, S. Mahmud, and U. Yousuf, "Big Data analytics and Computational Intelligence for Cyber-Physical Systems: Recent trends and state of the art applications," *Future Generation Computer Systems,* pp. 766–778, Nov 2017.

[21] T. M. Jones, "Ethical decision making by individuals in organizations: An issue-contingent model," *Academy of management review,* vol. 16, pp. 366-395, 1991.

[22] G. E. M. Anscombe, "Modern moral philosophy," *Philosophy,* vol. 33, pp. 1–19, 1958.

[23] I. G. Cohen, R. Amarasingham, A. Shah, B. Xie, and B. Lo, "The Legal And Ethical Concerns That Arise From Using Complex Predictive Analytics In Health Care," *Health Affairs,* vol. 33, pp. 1139–1147, Jul 2014.

[24] N. P. Shaw, A. Stöckel, R. W. Orr, T. F. Lidbetter, and R. Cohen, "Towards Provably Moral AI Agents in Bottom-up Learning Frameworks," *Aies '18,* pp. 271–277, 2018.

[25] A. Jobin, M. Ienca, and E. Vayena, "The global landscape of AI ethics guidelines," *Nature Machine Intelligence,* vol. 1, pp. 389-399, 2019.

[26] T. Hagendorff, "The Ethics of AI Ethics: An Evaluation of Guidelines," *Minds and Machines,* vol. 30, Mar 2020.

[27] M. Ryan and B. C. Stahl, "Artificial intelligence ethics guidelines for developers and users: clarifying their content and normative implications," *Journal of Information, Communication and Ethics in Society,* vol. 19, pp. 61-86, 2021.

[28] Y. Zhang, M. Wu, G. Y. Tian, G. Zhang, and J. Lu, "Ethics and privacy of artificial intelligence: Understandings from bibliometrics," *Knowledge-Based Systems,* vol. 222, p. 106994, 2021/06/21/ 2021.

[29] D. Moher, A. Liberati, J. Tetzlaff, and D. G. Altman, "Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement," *International Journal of Surgery,* vol. 8, pp. 336–341, Jan 2010.

[30] M. Wieringa, "What to account for when accounting for algorithms: a systematic literature review on algorithmic accountability," presented at the Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, 2020.

[31] A. Rossi and G. Lenzini, "Transparency by design in data-informed research: A collection of information design patterns," *Computer Law & Security Review,* vol. 37, pp. 1–22, Jul 2020.

[32] M. Büchi, E. Fosch-Villaronga, C. Lutz, A. Tamò-Larrieux, S. Velidi, and S. Viljoen, "The chilling effects of algorithmic profiling: Mapping the issues," *Computer Law & Security Review,* vol. 36, pp. 1–15, Apr 2020.

[33] E. Bertino, A. Kundu, and Z. Sura, "Data Transparency with Blockchain and AI Ethics," *Journal of Data and Information*

Quality, vol. 11, pp. 1–8, 2019.

[34] M. Loi, C. Heitz, and M. Christen, "A Comparative Assessment and Synthesis of Twenty Ethics Codes on AI and Big Data," in *2020 7th Swiss Conference on Data Science (SDS), 26-26 June 2020*, 2020, pp. 41-46.

[35] I. Munoko, H. L. Brown-Liburd, and M. Vasarhelyi, "The Ethical Implications of Using Artificial Intelligence in Auditing: JBE," *Journal of Business Ethics,* vol. 167, pp. 209-234, Nov 2020.

[36] T. Gebru, J. Morgenstern, B. Vecchione, J. W. Vaughan, H. Wallach, H. Daumé III*, et al.*, "Datasheets for datasets: arXiv preprint arXiv:1803.09010," *arXiv preprint arXiv:1803.09010,* 2018.

[37] R. Hamon, H. Junklewitz, G. Malgieri, P. De Hert, L. Beslay, and I. Sanchez, "Impossible Explanations? Beyond explainable AI in the GDPR from a COVID-19 use case scenario," presented at the Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event, Canada, 2021.

[38] B. Hutchinson, A. Smart, A. Hanna, E. Denton, C. Greer, O. Kjartansson*, et al.*, "Towards Accountability for Machine Learning Datasets: Practices from Software Engineering and Infrastructure," presented at the Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event, Canada, 2021.

[39] B. Wagner, K. Rozgonyi, M.-T. Sekwenz, J. Cobbe, and J. Singh, "Regulating transparency? Facebook, Twitter and the German Network Enforcement Act," presented at the Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, 2020.

[40] H. J. Watson and N. Conner, "Addressing the Growing Need for Algorithmic Transparency," *Communications of the Association for Information Systems,* vol. 45, p. 26, Mar 2019.

[41] J. Cobbe, M. S. A. Lee, and J. Singh, "Reviewable Automated Decision-Making: A Framework for Accountable Algorithmic Systems," presented at the Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event, Canada, 2021.

[42] O. H. Gandy, "Engaging rational discrimination: exploring reasons for placing regulatory constraints on decision support systems," *Ethics and Information Technology,* vol. 12, pp. 29–42, Mar 2010.

[43] J. H. Lim and H. Y. Kwon, "A Study on the Modeling of Major Factors for the Principles of AI Ethics," presented at the Digital Government Research (DG.O '21), June 09–11, 2021, Omaha, NE, USA, 2021.

[44] H. Adam, "The ghost in the legal machine: algorithmic governmentality, economy, and the practice of law," *Journal of Information, Communication and Ethics in Society,* vol. 16, pp. 16–31, 2018.

[45] J. Alasadi, A. A. Hilli, and V. K. Singh, "Toward Fairness in Face Matching Algorithms," presented at the Proceedings of the 1st International Workshop on Fairness, Accountability, and Transparency in MultiMedia, Nice, France, 2019.

[46] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? ," presented at the Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event, Canada, 2021.

[47] M. Eslami, K. Vaccaro, M. K. Lee, A. E. B. On, E. Gilbert, and K. Karahalios, "User Attitudes towards Algorithmic Opacity and Transparency in Online Reviewing Platforms," presented at the Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, Glasgow, Scotland Uk, 2019.

[48] M. Mitchell, S. Wu, A. Zaldivar, P. Barnes, L. Vasserman, B. Hutchinson*, et al.*, "Model Cards for Model Reporting," presented at the Proceedings of the Conference on Fairness, Accountability, and Transparency, Atlanta, GA, USA, 2019.

[49] M. Langer and R. N. Landers, "The future of artificial intelligence at work: A review on effects of decision automation and augmentation on workers targeted by algorithms and third-party observers," *Computers in Human Behavior,* vol. 123, p. 106878, 2021.

[50] A. R. Givens and M. R. Morris, "Centering disability perspectives in algorithmic fairness, accountability and transparency," presented at the Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, 2020.

[51] U.-V. Albrecht, "Transparency of Health-Apps for Trust and Decision Making," *Journal of Medical Internet Research,* vol. 15, pp. 1–5, Dec 2013.

[52] E. P. Vallejos, A. Koene, V. Portillo, L. Dowthwaite, and M. Cano, "Young People's Policy Recommendations on Algorithm Fairness," presented at the Proceedings of the 2017 ACM on Web Science Conference, Troy, New York, USA, 2017.

[53] J. Matthews, "Patterns and Antipatterns, Principles, and Pitfalls: Accountability and Transparency in Artificial Intelligence," *AI Magazine,* vol. 41, pp. 82-89, Nov 2020.

[54] U. Bhatt, A. Xiang, S. Sharma, A. Weller, A. Taly, Y. Jia*, et al.*, "Explainable Machine Learning in Deployment," presented at the Conference on Fairness, Accountability, and Transparency (Fat* '20), January 27–30, 2020, Barcelona, Spain, 2020.

[55] A. Mowbray, P. Chung, and G. Greenleaf, "Utilising AI in the legal assistance sector—Testing a role for legal information institutes," *Computer Law & Security Review,* vol. 38, pp. 1–9, Sep 2020.

[56] A. Joerin, M. Rauws, R. Fulmer, and V. Black, "Ethical Artificial Intelligence for Digital Health Organizations," *Cureus,* vol. 12, May 2020.

[57] B. Shneiderman, "Bridging the Gap Between Ethics and Practice: Guidelines for Reliable, Safe, and Trustworthy Human-Centered AI Systems," *ACM Trans. Interact. Intell. Syst.,* vol. 10, 2020.

[58] J. Metcalf, E. Moss, E. A. Watkins, R. Singh, and M. C. Elish, "Algorithmic Impact Assessments and Accountability: The Co-construction of Impacts," presented at the Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, 2021.

[59] A. Jacovi and Marasovi, "Formalizing Trust in Artificial Intelligence: Prerequisites, Causes and Goals of Human Trust in AI," presented at the ACM Conference on Fairness, Accountability, and Transparency (ACM FAccT '21), March 3–10, 2021, Virtual Event, Canada, 2021.

[60] A. Aguirre, G. Dempsey, H. Surden, and P. B. Reiner, "AI Loyalty: A New Paradigm for Aligning Stakeholder Interests," *IEEE Transactions on Technology and Society,* vol. 1, pp. 128-137, 2020.

[61] A. P. Brady and E. Neri, "Artificial Intelligence in Radiology—Ethical Considerations," *Diagnostics,* vol. 10, p. 231, Apr 2020.

[62] W. X. Wan and T. Lindenthal, "Towards Accountability in Machine Learning Applications: A System-testing Approach," *Available at SSRN,* 2021.

[63] G. Harrison, J. Hanson, C. Jacinto, J. Ramirez, and B. Ur, "An Empirical Study on the Perceived Fairness of Realistic, Imperfect Machine Learning Models," *Conference on Fairness, Accountability, and Transparency (Fat* '20), January 27–30, 2020,* pp. 392–402, 2020.

[64] S. K. McGrath and S. J. Whitty, "Accountability and responsibility defined," *International Journal of Managing Projects in Business,* vol. 11, pp. 687─707, Nov 2018.

[65] D. Rezania, R. Baker, and A. Nixon, "Exploring project managers' accountability," *International Journal of Managing Projects in Business,* vol. 12, pp. 919─937, Nov 2019.

[66] D. Shin and Y. J. Park, "Role of fairness, accountability, and transparency in algorithmic affordance," *Computers in Human Behavior,* vol. 98, pp. 277-284, Sep 2019.

[67] *Artificial Intelligence Act,* R. (EU) Proposal for a regulation of the European Parliament and of the Council: Laying down harmonised rules on artificial intelligence (Artificial Intelligence Act) and amending certain union legislative acts, 2021.

3rd Special Session on Data Science in Health, Ecology and Commerce

DATA Science in Health, Ecology and Commerce is a forum on all forms of data analysis, data economics, information systems and data based research, focusing on the interaction of those four fields. Here, data-driven solutions can be generated by understanding complex real-world (health) related problems, critical thinking and analytics to derive knowledge from (big) data. The past years have shown a forthcoming interest on innovative data technology and analytics solutions that link and utilize large amounts of data across individual digital ecosystems. First applications scenarios in the field of health, smart cities or agriculture merge data from various IoT devices, social media or application systems and demonstrate the great potential for gaining new insights, supporting decisions or providing smarter services. Together with inexpensive sensors and computing power we are ahead of a world that bases its decisions on data. However, we are only at the beginning of this journey and we need to further explore the required methods and technologies as well as the potential application fields and the impact on society and economy. This endeavor needs the knowledge of researchers from different fields applying diverse perspectives and using different methodological directions to find a way to grasp and fully understand the power and opportunities of data science.

This is a joint track by WIG2, the Scientific Institute for health economics and health service research, the Information Systems Institute of Leipzig University and the Helmholtz Environmental Research Institute.

TOPICS

We embrace a rich array of issues on data science and offer a platform for research from diverse methodological directions, including quantitative empirical research as well as qualitative contributions. We welcome research from a medical, technological, economic, political and societal perspective. The topics of interest therefore include but are not limited to:

- Data analysis in health, ecology and commerce
- (Health) Data management
- Health economics
- Data economics
- Data integration
- Semantic data analysis
- AI based data analysis
- Data based health service research
- Smart Service Engineering
- Integrating data in integrated care
- AI in integrated care
- Spatial health economics
- Risk adjustment and Predictive modelling
- Privacy in data science

TECHNICAL SESSION CHAIRS

- **Franczyk, Bogdan,** University of Leipzig, Germany
- **Militzer-Horstmann, Carsta,** WIG2 Institute for health economics and health service research, Leipzig, Germany
- **Häckl, Dennis,** WIG2 Institute for health economics and health service research, Leipzig, Germany
- **Bumberger, Jan,** Helmholtz-Centre for Environmental Research – UFZ, Germany
- **Reinhold, Olaf,** University of Leipzig / Social CRM Research Center, Germany

PROGRAM COMMITTEE

- **Alpkoçak, Adil,** Dokuz Eylul University
- **Cirqueira, Douglas,** Dublin City University
- **Dey, Nilanjan,** Techno India College of Technology, India
- **Kossack, Nils,** Head Mathematics and Statistics, WIG2 Institute for Health Economics and Health Service Research
- **Kozak, Karol,** Fraunhofer and Uniklinikum Dresden, Germany
- **Müller, Marco,** WIG2 Institute for Health Economics and Health Service Research
- **Popowski, Piotr,** Medical University of Gdańsk, Poland
- **Sachdeva, Shelly,** National Institute of Technology Delhi, India
- **Viehbahn, Malte,** WIG2 Institute for Health Economics and Health Service Research
- **Wasielewska-Michniewska, Katarzyna,** Systems Research Institute of the Polish Academy of Sciences, Poland

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 393–402

DOI: 10.15439/2021F78
ISSN 2300-5963 ACSIS, Vol. 25

Mass Vaccine Administration under Supply Uncertainty

Salvatore Foderaro
Università di Tor Vergata,
via del Politecnico 1, Roma, Italy
Email: salvatorefoderaro@gmail.com

Maurizio Naldi
Università LUMSA
via Marcantonio Colonna 19, Roma, Italy
Email: m.naldi@lumsa.it

Gaia Nicosia
Università Roma Tre,
via della Vasca Navale 79, Roma, Italy
Email: gaia.nicosia@uniroma3.it

Andrea Pacifici
Università di Tor Vergata,
via del Politecnico, 1, Roma, Italy
Email: andrea.pacifici@uniroma2.it

***Abstract*—The insurgence of COVID-19 requires fast mass vaccination, hampered by scarce availability and uncertain supply of vaccine doses and a tight schedule for boosters. In this paper, we analyze planning strategies for the vaccination campaign to vaccinate as many people as possible while meeting the booster schedule. We compare a conservative strategy and q-days-ahead strategies against the clairvoyant strategy. The conservative strategy achieves the best trade-off between utilization and compliance with the booster schedule. Q-days-ahead strategies with $q < 7$ provide a larger utilization but run out of stock in over 30% of days.**

I. Introduction

DUE to the global COVID-19 pandemic emergency, mass vaccinations are taking place all over the world. Mass vaccinations have been held in the past, starting with the vaccination days and the mass campaign to eradicate smallpox in the early 19th century [1]. However, two centuries after, mass vaccination is still a challenge [2]. The challenge is particularly severe for situations where the need for mass vaccination arises while vaccines are being developed. In that case, time constraints conflict: herd immunity calls for fast action, but the need for wide vaccine availability slows down the campaign deployment. As a consequence, a great variety of logistic problems connected to this enormous task arise. In this paper, we focus on planning a vaccination campaign, i.e, determining a day-by-day prescription on the number of doses that have to be administered in the presence of uncertainties in the distribution provided by the vaccine suppliers. Here, we are not looking at allocation options concerning the priority order of specific population segments or areas in the territory. Instead, we focus on the *downstream* problem connected to the effective and efficient delivery of the vaccines to eligible individuals. In designing such a decision support tool, several issues must be considered, mainly due to different characteristics of the administered vaccines.

- Inventory Management issues: vaccine products are temperature-sensitive and must be stored and handled correctly to ensure efficacy and maximize shelf life. Proper storage and handling practices are critical to minimize vaccine loss and limit the risk of administering the COVID-19 vaccine with reduced effectiveness. Expiration dates have also to be taken into account. It is also to be noted that timely management of inventories requires the fast updating and integration of hospital information systems [3] as well as their reliability [4].
- Booster shot: whether or not a second dose is required and the prescribed time interval between the first and the second doses is a component that greatly affects any planning model for mass vaccination.
- Overall vaccination capacity: we must consider the maximum number of vaccinations (independently of whether they are first or second doses) that the system can administer every day. We are considering this information as given and deterministic, to be set as a function of the number of operators and vaccination sites capacity;
- Trade-off between different procurement and vaccination strategies under a limited budget and time horizon constraints, which we do not consider here, but should enter the more general analysis framework [5].

The paper is organized as follows.

Hereafter we briefly report on a few related works in the literature and describe the context and notation for the addressed problem.

In Section IV, we illustrate how the arrival process is modelled and, consequently, how the input data of the experiments are generated.

Optimization models and approaches to the problem are presented in Section V, where two main points of view are considered. We first address the problem as if it were a deterministic (*off-line*) one, i.e., all data are assumed to be known and given in advance. The output of such a phase is a point of reference or benchmark to assess the quality of different non-clairvoyant methods presented in the same

This work is partially supported by MIUR PRIN Project AHeAD (Efficient Algorithms for HArnessing Networked Data).

section. We then consider models where supply is regarded as a random process over time, are designed to tackle the real-world stochastic problem effectively.

Section VI reports the results of an extensive computational campaign aimed at testing and assessing the effectiveness of the different proposed approaches.

Finally, in Section VII, some conclusions are drawn.

II. Related literature

Due to the current pandemic situation, mass vaccination logistics problems came (overwhelmingly indeed!) to the attention of researchers only very recently. As a consequence, the literature concerning this specific area is still relatively scarce. There are, however, several papers dealing with various problems arising in the event of a sudden burst of infections caused by a pathogen in a population.

A comprehensive introduction to the mathematical modelling of infectious diseases can be found in the book by Keeling and Rohani [6] as an essential tool in public health planning and response. Several techniques are illustrated to model basic epidemiological processes, such as the propagation of infectious diseases. Such techniques range from differential equations to computer simulations.

The effectiveness of mass vaccination against other policies (such as trace vaccination) is discussed in [7] for the hypothetical case of a smallpox bioterrorist attack in a large U.S. city.

Among the few papers dealing with mass vaccination logistics in the most recent literature (not necessarily related to the COVID-19 pandemic), the following ones present some appreciable connections with the problem at hand.

In [8] the authors formulate a bi-objective model for planning vaccination campaigns that aim at minimising both control costs and the number of infected individuals.

In the context of mass vaccination, some papers in the literature have addressed different types of problems. In a very recent paper, the authors address the problem of allocating vaccines across geographical regions to utilise available vaccines as effectively as possible [9].

Both the above papers base their analysis on the epidemiological conditions of a population and/or a geographical area.

Unlike the above studies, our aim in this work is not to identify who should get vaccinated first. Instead, we want to establish how to optimally exploit the uncertain supplies of doses to speed up the vaccination process and rapidly immunise the largest possible fraction of the population.

In fact, the problem we address here resembles the so-called lot-sizing in production planning, which has been extensively studied since the seminal work of Wagner and Whitin in the Fifties [10]. In [10], the authors propose a forward dynamic programming algorithm for a generalised version of the uncapacitated economic lot-sizing model with dynamic demand under a general concave cost function. The latter model has been extended in [11] by considering the possibility of backlogging. These prototypical models can be viewed as special fixed charge network problems. Several variants have been investigated and still are an important topic of research, including single-item and multi-item, uncapacitated and capacitated lot-sizing problems. However, differently from lot-sizing problems, as we are discussing below, in our model, we are not considering inventory costs as a main component of the decision criteria. Though stock expenses are not negligible, we prioritise the average vaccination time since a fast immunisation of the largest possible audience is of much greater importance in this situation.

III. Context and problem definition

The purpose of our study is to design a support tool providing the decision-maker with a suggestion about the number of doses of vaccines to be administered every day along a given planning horizon. The set of vaccine types is $\mathcal{V}$, and the planning horizon consists of a number T of periods (days):

$$\mathcal{W} = \{1, \dots, T\}.$$

This information specifies the type of vaccine, a first or a booster dose, and the daily inventory level for each vaccine. These decisions are based on imperfect information concerning the supply of doses during the planning time window: The number b_t^i of doses of vaccine $i \in \mathcal{V}$ delivered at day $t \in \mathcal{W}$ is considered as a random variable and, in Section IV, the arrival process is thoroughly described using suitable probability density functions.

We wish to (i) determine benchmarks against which the algorithms for the non-deterministic case can be confronted, and (ii) to design a tool that can be safely used when the information about the arrivals will be more reliable, as it can be expected in the (hopefully near) future. For those reasons, in Section V-A, we are also considering an *off-line* version of the problem, in which the amount of daily provision b_t^i for each type of vaccine are given as deterministic data.

Another critical input parameter is the capacity limit of the system, i.e., the maximum number of vaccine doses that may be administered at time t. This limit could be independent of the administered vaccine type. However, in our algorithms, we are considering upper bounds k_t^i on the number of each single vaccine type i that can be administered in day t as given input, for all $i \in \mathcal{V},\ t \in \mathcal{W}$. Of course, the size of these variables (so that $\sum_{i\in\mathcal{V}} k_t^i$ is a constant or slightly variable over time) can be suitably tuned, depending on the available supply.

The algorithms we are proposing are basically models that return a prescriptive vaccination plan over the next T days. This is especially true of off-line algorithms. In particular, the output of the algorithm is the number x_t^i, resp. y_t^i, of people receiving the first, resp. the second, dose of vaccine i on day t. As a consequence, an additional output of these procedures is the stock level of each vaccine i at (the end of) day $t \in \mathcal{W}$. Clearly, the algorithms could also be used in a *rolling-horizon'* fashion, i.e., re-optimizing every single or one-in-n day, taking into account current inventory levels and new estimated future dose arrivals.

For the developments to follow, we refer to the total number of supplied doses of vaccine i until day t by:

$$B^i(t) = \sum_{\theta=1}^{t} b^i_\theta$$

Observe that, if Δ^i is the recommended time interval, expressed in number of periods, between the first and the mandatory booster (or second) dose of vaccine $i \in \mathcal{V}$, since $\sum_{t=1}^{T-\Delta^i} x^i_t = \sum_{t=\Delta^i+1}^{T} y^i_t \le \frac{1}{2}B^i(T)$ and $\sum_{t=1}^{T-\Delta^i} x^i_t \le B^i(T-\Delta^i)$, then there is a feasible solution with $s^i_T = 0$ if and only if $B^i(T-\Delta^i) \ge \frac{1}{2}B^i(T)$. In fact, any feasible solution has $s^i_T \ge B^i(T) - 2B^i(T-\Delta^i)$. (With no loss of generality, we assume that the right-hand side of the latter inequality is not positive. Otherwise, we may subtract this quantity from the arrivals of the last $T-\Delta^i$ days and then apply the algorithm. So doing we eventually have $s^i_T = B^i(T) - 2B^i(T-\Delta^i)$.)

IV. The Arrival Process

Vaccine administration is fed by the availability of vaccine doses. A smooth and regular procurement and delivery process of vaccine doses (and the vaccine supply chain in general) is essential to the correct planning of the administration phase [12], [13]. However, the delivery of doses has been hampered by repeated delays, well reported in the general press [1]. As a consequence, the delivery of doses to nations, and subsequently to vaccination centers, appears as largely random. In this section, we provide a stochastic model for the arrival of vaccine doses, considering a whole nation as the recipient.

For this purpose, we rely on the datasets provided for Italy under an OpenData agreement at https://github.com/italia/covid19-opendata-vaccini. The datasets are updated daily and include the number of doses received for each supplier (which are AstraZeneca, Moderna, Pfizer-BioNTech, and Johnson&Johnson). For the time being, the observation interval considered for this study is from December 27, 2020, to April 2, 2021. The sample record of arrivals for Pfizer-BioNTech is shown in Fig. 1. We can notice two major features of the time series:

- the arrivals exhibit a growing trend;
- arrivals do not take place each day, and on most days, there are no arrivals.

The first feature is probably a consequence of the current transient nature of the process. Pharmaceutical companies are scaling up their production to meet the growing demand of nations to vaccinate their citizens. As seen at this stage, the resulting stochastic process of arrivals would be non-stationary, calling, e.g., for the use of a non-homogeneous Poisson model [14]. However, we are more interested in the steady-state version of the process since we imagine an ongoing massive vaccination after facing today's initial phase.

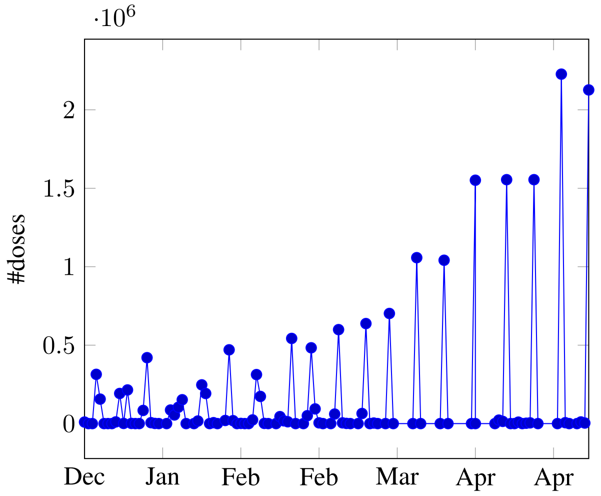


Fig. 1. Daily arrivals of Pfizer-BioNTech doses over Dec 2020 - May 2021

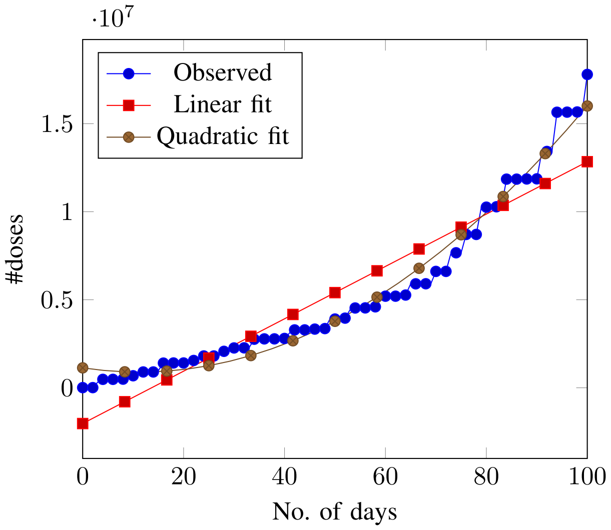


Fig. 2. Cumulative arrivals of Pfizer-BioNTech doses (Dec 2020 - May 2021)

We can, however, examine the growing transient to compare a homogeneous against a non-homogeneous Poisson process (as adopted, e.g. in [15]). In Fig. 2, we have reported the cumulative number of dose arrivals. If that process followed a homogeneous Poisson model, the expected number of arrivals over any period would be proportional to that period (i.e., linear in time). On the same Fig. 2, we have also reported the linear trend that would better fit the observed data. As we can see, the linear trend is a poor approximation of the real growth of arrivals. A quadratic function, also shown in the picture, would be a better fit.

As to the second feature, delivery days are interspersed with relatively long intervals of no-delivery. For that reason, we are led towards a zero-inflated model, where zeros occur more often. In zero-inflated models, the occurrence of zero arrivals is superimposed with a model assuming a larger domain (in our case, the domain $\mathbf{N}$) [16], [17]. The latter

[1] See, e.g., just the recent headlines “Covid vaccine: UK supply hit by India delivery delay” at https://www.bbc.com/news/uk-56438629 and “Covid: What is happening with the EU vaccine rollout?” at https://www.bbc.com/news/explainers-52380823.

model may be a Poisson, negative binomial, binomial, beta-binomial or hypergeometric. Here we have opted for a zero-inflated Poisson model, also known as ZIP. We recall that we are considering the steady-state scenario with a homogeneous ZIP model. Should we wish to examine the transient phenomenon, we could consider a quadratic approximation for the expected number of arrivals in a non-homogeneous zero-inflated Poisson model, as shown earlier.

In the ZIP model, the probability distribution for the number X of dose arrivals is

$$\mathbf{P}[X=k] = \left\{ \begin{array}{ll} \pi+(1-\pi)e^{-\lambda} & \text{if } k=0, \\ (1-\pi)e^{-\lambda}\lambda^k/k! & \text{if } k=1,2,\dots. \end{array} \right. \tag{1}$$

where $0 \leq \pi \leq 1$ and $\lambda \geq 0$.

This model is then represented by two parameters, π and λ. We can adopt several methods to estimate those parameters [18]. In particular, if we indicate the sampling mean and variance respectively as m and s^2, the method of moments provides us with the following estimates:

$$\begin{aligned} \hat{\pi} &= \frac{s^2-m}{s^2+m^2-m} \\ \hat{\lambda} &= \frac{s^2+m^2}{m} - 1. \end{aligned} \tag{2}$$

V. Vaccination Planning Approaches

In this section, we describe the algorithms that can be used as a decision support tool for designing a vaccination plan over the next T days. The plan consists of establishing the number of (first and second) doses of vaccines that shall be administered every day.

A. *Off-line optimization model*

Hereafter, we present a linear programming model providing a solution to maximize the number of vaccinated people per day while exploiting all the doses supplied during the planning time window $\mathcal{W} = \{1, \dots, T\}$.

The set of available vaccine types is $\mathcal{V} = \mathcal{V}' \cup \mathcal{V}''$. Vaccine type $\mathcal{V}'$ requires a single dose while vaccine type $\mathcal{V}''$ requires a double shot.

In our model, we consider the following decision variables:

- x_t^i and y_t^i, $t \in \mathcal{W}$, $i \in \mathcal{V}$ indicate the number of first and second doses of vaccine i administered during time t;
- s_t^i is the amount of doses of vaccine i remaining in stock at (the end) of period t.

We also assume that, in the initial period $t = 1$, a given inventory $s_0^i \geq 0$ of doses is available in stock and that a given maximum inventory level $\bar{s}_F^i \geq 0$ is required at the end of the planning horizon. We discuss possible feasible values for $\bar{s}_F^i$ later on.

The LP model is

$$\begin{array}{lll} \min f(y) & & (3) \\ s.t.\ \ x_t^i + y_t^i + s_t^i - s_{t-1}^i = b_t^i & i \in \mathcal{V}, t \in \mathcal{W} & (4) \\ \quad x_t^i + y_t^i \leq k_t^i & i \in \mathcal{V}, t \in \mathcal{W} & (5) \\ \quad x_t^i = y_{t+\Delta^i}^i & i \in \mathcal{V}'', t \in \mathcal{W} & (6) \\ \quad x_t^i = 0 & i \in \mathcal{V}', t \in \mathcal{W} & (7) \\ \quad s_T^i \leq \bar{s}_F^i & i \in \mathcal{V}, t \in \mathcal{W} & (8) \\ \quad x_t^i, y_t^i, s_t^i \geq 0 & i \in \mathcal{V}, t \in \mathcal{W} & (9) \end{array}$$

In this model the objective function (3) is the average vaccination time that may be expressed as

$$f(y) = \frac{\sum_{t \in \mathcal{W}} (t \sum_{i \in \mathcal{V}} y_t^i)}{\sum_{t \in \mathcal{W}} \sum_{i \in \mathcal{V}} y_t^i} \tag{10}$$

Note that, each time period t is "weighted" by the number of people $\sum_i y_t^i$ receiving the final dose at t. Equation (10) can be linearized by approximating the denominator with $\frac{1}{2}\sum_{i \in \mathcal{V}} B^i(T)$, i.e., the maximum number of second doses that can be administered in the face of certain dose-supplies b^i: In presence of null final stocks $s_T^i = 0$, the two expressions have equal values.

Equations (4) are simple continuity constraints expressing the obvious relationship among the variables and the supplied number of doses. Constraints (5) bound the total number of doses administered in each period. Constraints (6) ensure that the second dose of the vaccination is given after the recommended time interval, while constraints (7) refer to the single-dose vaccines[2] Constraints (8) impose that the final inventory level is at most a given quantity $\bar{s}_F^i$. Without such constraints, the optimal solution would be not to administer any vaccine (and obtain a null valued objective). The values $\bar{s}_F^i$ are given as an input to the LP model and can be chosen small enough to guarantee that we are using as much as possible of the supplied doses. In any case, it is clear that

$$\bar{s}_F^i \geq \max\{0, B^i(T) - 2B^i(T - \Delta^i)\} \tag{11}$$

should hold.

The above model computes an optimal solution of our planning problem under the assumption that the exact amount of supply b_t^i of each vaccine i is given, for each period t, i.e., the LP solves a deterministic (off-line) version of the actual stochastic problem. Such solutions can be used as a benchmark to assess the quality of blind heuristic algorithms providing prescriptive information on the number of doses that can be administered each day without a perfect knowledge on the doses supplied in the future. As it is expected that the supply process will become steady and the data about arrival dates trustworthy enough, the above linear programming models could be used as a reliable decision support system.

[2] While we show that single-dose vaccines may be easily included in our models, due to scarcity of data about this type of immunization, in the remainder of the paper we only present algorithms and experiments concerning two-doses vaccines.

With regard to the off-line version of the problem, in which the supply b_t^i is deterministic and given for all $i \in \mathcal{V}, t \in \mathcal{W}$, one may ask if a simpler mechanism than the LP-based one described above would determine an optimal (or close to optimal) solution without recurring to a mathematical program. A greedy strategy seems a viable tool due to the simple continuity relations binding the different quantities together (similar, for instance, to those of the classical lot-sizing problem). Therefore, limiting ourselves to the uncapacitated case, we tested a natural heuristic that guarantees the maximum consumption of all the supplied doses and tries to schedule as early as possible the vaccinations while keeping inventory levels nonnegative. Note that the problem is decomposable, and the heuristic provides a separate administration plan for each vaccine $i \in \mathcal{V}$. Such a heuristic algorithm is sketched hereafter.

We first define an initial feasible solution $\bar{x}$ as:

$$\bar{x}_t^i = \begin{cases} \frac{1}{2}B^i(T) & t = T - \Delta^i \\ 0 & t \in \mathcal{W} \setminus \{T - \Delta^i\} \end{cases} \tag{12}$$

This is a feasible solution in which all the first [second] doses are administered in the last possible time slot, namely $T - \Delta^i$ [T]. Moreover, no stock is left at the end of the planning horizon, i.e., $s_T^i = 0$.

Starting from $\bar{x}^i$, our algorithm tries to move vaccinations earlier in a greedy fashion while always keeping the following relation true

$$x_t^i + y_t^i \leq B^i(t). \tag{13}$$

Algorithm 1 Greedy off-line heuristic

```
for all $i \in \mathcal{V}$ do
    $x^i := \bar{x}^i$;
    for $t = T - \Delta^i$ down to 2 do
        $c := \min\{s(t + \Delta^i - 1), x_t, B(t-1)\}$;
        Augment $x_{t-1}^i$ and $y_{t+\Delta^i-1}^i$ by $c$;
        Decrease $x_t^i$, $y_{t+\Delta^i}^i$, $s_{t-1}^i$ and $s_{t+\Delta^i-1}^i$ by $c$
    end for
    return $x^i$
end for
```

It is not hard to see that, at each step, c is the maximum amount of doses that can be moved earlier without violating the constraint $s_t \geq 0$.

In Section VI, we will report about the performance of the above described Algorithm 1.

B. Blind Algorithms

Hereafter we illustrate different greedy approaches to the decision on the number of doses to administer each day t, when no clairvoyance can be assumed on the future arrivals of doses. Note, however, that the plan output at t exploits the knowledge of the supply b_t^i on that day, for each vaccine i.

A simple idea consists in imposing that the amount of stock at the end of each day has to be equal at least to the number of second doses to be administered the next day (day-by-day no-out-of-stock condition).

The balance equation at the end of day t for the i-th vaccine is

$$\begin{aligned} s_t^i &= s_{t-1}^i - x_t^i - y_t^i + b_t^i \\ &= s_{t-1}^i - x_t^i - x_{t-\Delta^i}^i + b_t^i, \end{aligned} \tag{14}$$

due to the delay introduced between the first and second dose.

If we do not want to run out of stock and guarantee the administration of the second dose at day $t+1$, we must impose the following no-out-of-stock condition, which assumes that no doses arrive on day $t+1$ is equivalent to the following relation

$$s_t^i \geq y_{t+1}^i = x_{t+1-\Delta^i}^i. \tag{15}$$

Equation (15) embodies the day-by-day administration strategy. Since new arrivals may arrive on day t the number of first doses that can be safely administered at time t while still meeting the condition of Equation (15) is

$$x_t^i \leq s_t^i - x_{t-\Delta^i}^i - x_{t+1-\Delta^i}^i + b_t^i. \tag{16}$$

Again, in Equation (16), three quantities are known at the end of time $t-1$ (s_{t-1}^i, $x_{t-\Delta^i}^i$, and $x_{t+1-\Delta^i}$), while the fourth one (b_t^i) is random but will be known at the beginning of day t.

If the sum in the right-hand term of Equation (16) is not positive, then we will not be able to administer any first dose on day t. This unfortunate situation takes place if the number of arrivals on day t is

$$b_t^i \leq z_t^i = x_{t-\Delta^i}^i + x_{t+1-\Delta^i} - s_{t-1}^i \tag{17}$$

The risk of incurring the no-first-doses situation is then $\mathbb{P}[b_t^i \leq \gamma_t^i]$. For the ZIP process, this risk $\mathbb{R}_{\text{no-1}}^{(1)}$ is

$$\mathbb{R}_{\text{no-1}}^{(1)} = \begin{cases} 0 & \text{if } z_t^i < 0, \\ \pi + (1-\pi)\sum_{i=0}^{\gamma_t^i} \frac{\lambda^i}{i!} e^{-\lambda} & \text{if } \gamma_t^i \geq 0 \end{cases} \tag{18}$$

On the other hand, if no arrivals take place for a long period of time, we cannot even guarantee that the day-by-day strategy is applicable. In that case, we would not even be able to administer all the second doses. This very unfortunate case takes place if the following condition holds

$$s_{t-1}^i + b_t^i < y_t^i \rightarrow b_t^i < \eta_t^i = y_t^i - s_{t-1}^i \tag{19}$$

The risk of incurring the no-second-doses situation is then $\mathbb{P}[b_t^i \leq \eta_t^i]$. For the ZIP process, this risk $\mathbb{R}_{\text{no-2}}^{(1)}$ is

$$\mathbb{R}_{\text{no-1}} = \begin{cases} 0 & \text{if } \eta_t^i < 0, \\ \pi + (1-\pi)\sum_{i=0}^{\eta_t^i} \frac{\lambda^i}{i!} e^{-\lambda} & \text{if } \eta_t^i \geq 0 \end{cases} \tag{20}$$

Extending the above arguments, we may derive a prescription on the amount of first doses to be safely administered so that we are guaranteed that the stock we have at time $t-1$ is enough to cover the overall demand over the next q days, establishing a sliding window that is shifted each day. This more conservative administration strategy may be called the q-days-ahead strategy (it is readily seen that the day-by-day strategy represents a special case of the q-days-ahead one when $q = 1$.)

In this case, the no-out-of-stock condition (15) clearly becomes

$$s_t^i \geq \sum_{\ell=1}^{q} y_{t+\ell}^i \tag{21}$$

As a consequence, if we include capacity constraints on the maximum number of doses that can be administered per day, the number of first doses that can be safely administered at time t would satisfy the following inequality:

$$x_t^i \leq \max\{0, \min\{k_t - y_t^i, s_{t-1}^i - \sum_{\ell=1}^{q} y_{t+\ell}^i + b_t^i\}\}. \tag{22}$$

The above relation directly suggests a simple myopic algorithm in which the risk of running out-of-stock is inversely proportional to the value of q.

Since we can rely on new arrivals to meet the constraint on the stock at the end of the day, we will not be able to safely administer first doses if the following condition holds

$$\begin{aligned} s_{t-1}^i - x_{t-\Delta^i}^i + b_t^i - \sum_{\ell=1}^{q} y_{t+\ell}^i \leq 0 \rightarrow \\ b_t^i \leq \phi_t^i = \sum_{\ell=0}^{q} y_{t+\ell}^i + x_{t-\Delta^i}^i - s_{t-1}^i. \end{aligned} \tag{23}$$

In the case of the ZIP process, the risk of not being able to administer any first dose is then

$$\mathbb{R}^{(q)}_{\text{no-1}} = \begin{cases} 0 & \text{if } \phi_t^i < 0, \\ \pi + (1-\pi)\sum_{i=0}^{\phi_t^i} \frac{\lambda^i}{i!} e^{-\lambda} & \text{if } \phi_t^i \geq 0 \end{cases} \tag{24}$$

Similarly, the risk of not even meeting the full demand for second doses equals that of not satisfying the inequality (19). Though the resulting risk expression equals that expressed for the day-by-day strategy in Equation (20), we must consider that the condition applied for the q-days-ahead strategy in Equation (21) clearly includes the one for the day-by-day strategy in Equation (15). The conclusion is that the risk of not being able to administer either first or second doses in the q-days-ahead strategy is surely lower than that suffered in the day-by-day strategy.

Though a suitable choice of q avoids the risk of dose shortages with a reasonable degree of confidence, it does not rule out the possibility of such an undesirable situation. As a consequence, in our experiments, we will consider (and assess the performance) of a *conservative* algorithm that guarantees that an adequate level of inventory is always available for second doses — with no knowledge on the future supplies. As one may expect, this conservative attitude of the algorithm has, of course, its disadvantages in terms of residual inventory and average vaccination times.

The basic idea is that whenever a number b of doses becomes available at t, one can immediately administer $b/2$ first doses and reserve the remainder to administer the corresponding second doses after the prescribed period. The output plan is obtained by augmenting the current solution each time a positive supply of new doses arrives. If the capacity (at t or $t + \Delta^i$) limits the number of doses that could be administered now, the possible excess of available doses at t gets transferred to the next period, and the procedure is iterated.

The conservative Algorithm 2 is illustrated hereafter. For each vaccine i and day t, we store the amounts x_t^i of the first doses to be administered. The current amount of supply available for the first doses is stored in a_t^i. Moreover, the parameter k_t^i is an upper bound set on the number of doses of vaccine i (capacity) that can be administered on the day t.

Algorithm 2 Conservative algorithm

```
1: for $i \in \mathcal{V}$ do
2:     Initialize $x$, $y$ and $a$ as null vectors;
3:     for $t = 1, 2, \ldots, T$ do
4:         $a_t^i := a_t^i + b_t^i$;
5:         $\delta := \max\{x_t^i + y_t^i + \frac{1}{2}a_t^i - k_t^i,\ x_{t+\Delta^i}^i + y_{t+\Delta^i}^i + \frac{1}{2}a_t^i - k_{t+\Delta^i}^i\}$; {Excess is computed}
6:         if $\delta > 0$ then
7:             $a_{t+1}^i := a_{t+1}^i + \frac{1}{2}a_t^i + \delta$;
8:             $a_t^i := a_t^i - 2\delta$ {Excess is transferred}
9:         end if
10:        if $a_t^i > 0$ then
11:            $x_t^i := x_t^i + \frac{1}{2}a_t^i$;
12:            $y_{t+\Delta^i}^i := y_{t+\Delta^i}^i + \frac{1}{2}a_t^i$
13:        end if
14:    end for
15: end for
```

The idea is to compute the excess availability with respect to the capacity values for days t and $t + \Delta^i$. The quantity δ represent such an excess: If it is positive, we are not allowed to administer all the available $a_t^i/2$ doses plus the previously planned amount of $x_t^i + y_t^i$ vaccines. In this case, we are reducing the availability at the current day t and reserve the exceeding quota of vaccines as available for the next day $t+1$ (Steps 7 and 8.)

Note that, when $\delta > 0$, the maximum number of *additional* doses that can be administered at day t is the minimum between $k_t^i - x_t^i - y_t^i$ and $k_{t+\Delta^i}^i - x_{t+\Delta^i}^i - y_{t+\Delta^i}^i$. One can easily see that such a minimum is given by $\frac{1}{2}a_t^i - \delta$ and hence that the actual excess is $\frac{1}{2}a_t^i + \delta$. The latter quantity is therefore added to the availability a_{t+1}^i of the next day while it is cut from a_t^i. (For instance, if $\delta = x_t^i + y_t^i + \frac{1}{2}a_t^i - k_t^i > 0$, then a_t^i becomes $a_t^i - 2\delta$ and x_t^i is augmented to the maximum value allowed by the capcity limit, i.e., $x_t^i := x_t^i + \frac{1}{2}a_t^i = k_t^i - y_t^i$. Similarly, if $\delta = x_{t+\Delta^i}^i + y_{t+\Delta^i}^i + \frac{1}{2}a_t^i - k_{t+\Delta^i}^i > 0$ then we obtain that $y_{t+\Delta^i}^i = k_{t+\Delta^i}^i - x_{t+\Delta^i}^i$.)

VI. Experiments and results

In this section, we describe our computational experience to compare the different approaches described above that design daily vaccination plans over a time horizon of T days. We first define a set of performance metrics and then report the results of our simulation experiments.

As a model for the arrival process, we have employed a ZIP model where the parameters are respectively $\pi = 0.85$ and $\lambda = 10^7$. Those parameters give us an average daily number of doses equal to $1.5 \cdot 10^6$, which is consistent with the current trend in Europe. In Fig. 4, we see that European countries currently lie in the 0.5-1 million bracket, but are following a growing trend[3]. We have run 1000 simulations to generate as many realistic arrival scenarios, each corresponding to an instance of the problem. The capacity is set as a multiple (through the capacity factor) of the average number of arrivals.

In the first set of experiments, we compared the performance of the linear program (3)-(9) against the greedy heuristic Algorithm 1. The empirical probability density function of the relative gap $\frac{H-LP}{H}\%$ between the objective function values LP and H obtained by the linear program and the heuristic, respectively, is shown in Fig. 3. We have employed the Gaussian kernel method [19], with a data-driven kernel bandwidth set as in [20], equal to 0.3719 in our case. As can be seen, the mode is around 2%. As discussed above, both algorithms provide an off-line benchmark to measure the performance of the blind algorithms. The tests show that the heuristic algorithm is quite effective in most cases: The average gap is 2.57% with more than 90% of the instances with a relative gap below 5%. Though the greedy heuristic finds a solution much faster than the linear programming solver and we need to compute an optimal off-line solution several times, it is still definitely compatible with our experiments to use the optima provided by the linear programming solver, as, on the average, an optimum of a single instance is computed by Gurobi in around 60 milliseconds. (Indeed, running all five heuristics on a single instance requires around six milliseconds.) It is clear that in a different context, e.g., larger instances (greater T values), a larger number of scenarios, additional constraints, etc., due to the effectiveness of the greedy heuristic witnessed by the low gap values, the proposed algorithm might be a good alternative approach to provide the necessary off-line benchmarks.

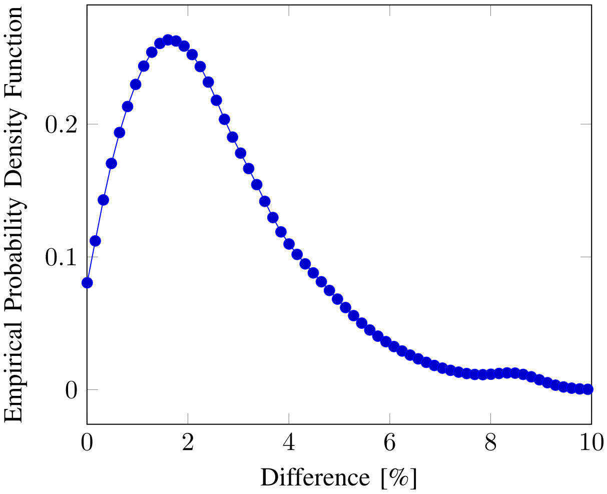


Fig. 3. Distribution of objective function percentual gap for Algorithm 1.

Recall that the capacity values k_t^i measure the maximum number of doses of vaccine i that the system is able to dispense in day t, for all $i \in \mathcal{W}$. In our tests, for each arrival scenario, we have considered 14 runs of the algorithms each corresponding to a specific capacity value:

$$k_t^i = (1 + \frac{\alpha}{2})c^i \quad \alpha = 0, 1, 2, \ldots, 13; \; i \in \mathcal{V}; \; t \in \mathcal{W} \tag{25}$$

in which, for all $i \in \mathcal{V}$, $c^i = \frac{1}{t}\sum_{t\in\mathcal{W}} b_t^i$ is a base-step capacity[4] (corresponding to the number of average per day arrival for vaccine i) and $(1 + \frac{\alpha}{2})$ is a *capacity factor*.

All the experiments here reported were run on a PC with CPU Intel $i5 - 5300U$ 64bit 2.30GHz clock, and 8GB RAM. The algorithms were coded in Python 3, v3.8.5. The implementation of the LP model solver makes use of the Python Gurobipy library 9.1.2 [21].

In our experiments we measure the following metrics:

- *Average vaccination time*. When $s_T^i = 0$ the average vaccination time is exactly equal to the expression given in Equation (10). In order to compare the results among experiments in which there are different levels of unused stocks at the end of the planning period, we use the following expression:

$$\frac{\sum_{i\in\mathcal{V}}(s_T^i(T + \Delta^i) + 2\sum_{t\in\mathcal{W}} t y_t^i)}{\sum_{i\in\mathcal{V}}\sum_{t\in\mathcal{W}} b_t^i}. \tag{26}$$

  As in Equation (10), the denominator counts (twice) the total number of administered vaccines. The numerator is the sum of the vaccination days $t = 1, \ldots, T$, each weighted by (twice) the number of second doses administered that day. The additive term $s_T^i(T + \Delta^i)$ accounts for those doses that remain unused at the end of the day T and are then stocked for the next days. Basically, we are considering that half of those doses will be administered on the same day T and, as booster vaccinations, after Δ^i days.
- *Utilization*. It measures how efficiently the supplied doses have been utilized at the end of the planning period: This figure is simply the ratio between the total number of planned final doses $\sum_i \sum_t y_t^i$ over *half* of the total number of supplied doses $B(T)$ (which is the theoretical maximum). Clearly, low levels of utilization mean that at the end of the planning period, the proposed approach has in stock a significant number of doses. It is important to stress that these data include possible backlogs in the planned number of delivered vaccines. That is, a positive y_t^i value may exist in correspondence to a negative value of inventory level s_t^i. In such a case, this is equivalent to assume that s_t^i vaccines would take place on a day later

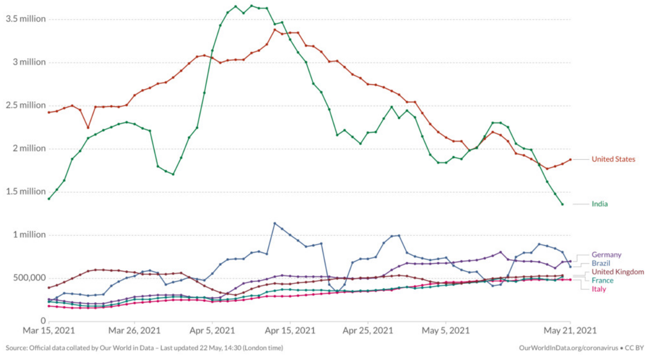
[3] See the latest data published on https://ourworldindata.org/grapher/daily-covid-19-vaccination-doses

[4] The idea is that a system with a base-step capacity c^i for all $i \in \mathcal{V}$ would be able to consume all the arrived doses of vaccine i, at the end of the planning period, only if it would deliver doses at its capacity level, each single day.

than t, the planned day. In this regard, we also report the following two figures.

- *Number of out-of-stock days.* The number of days in which there would be a shortage of doses, i.e., $s_t^i < 0$. In the graphics, we report these numbers as a percentage over the number T of days of the planning period.
- *Average backlog.* It is the absolute value of the average negative stocks $\frac{1}{T}\sum_i \sum_t |\min\{0, s_t^i\}|$ and it is also a measure of the average number of doses that would be administered after their prescribed date. This quantity might be of help in sizing an adequate level of safety stocks. For ease of readability, in the graphic of Fig. 7 this measure is reported as a percentage over the total number of supplied doses:

$$\frac{\sum_i \sum_t |\min\{0, s_t^i\}|}{T \sum_i \sum_t b_t^i}\%.$$



Source: Official data collated by Our World in Data – Last updated 22 May, 14:30 (London time)

OurWorldInData.org/coronavirus • CC BY

Fig. 4. Daily vaccination rates

We first take a look at the average vaccination time in Fig. 5. Of course, we aim at the lowest vaccination time possible. We see that the clairvoyant strategy achieves the best performance, as expected: the average vaccination time is roughly four months away from the start of the vaccination campaign. All the look-ahead strategies perform worse, with times getting longer as we lengthen our look-ahead horizon, playing safe against long periods of no dose arrivals. While guaranteeing that all second doses are administered on time, the conservative approach achieves an average vaccination time, which is not the worst in the group, ranking between the 7-days-ahead and the 14-days-ahead strategy. All the curves flatten out as the capacity factor grows. It appears then useless to have a capacity factor larger than 2 in most cases.

While both the clairvoyant and the conservative strategy guarantee that all second doses needs are met on the very same day, that's not the case for other strategies. We see that the vaccination system may incur a dose debt, where vaccination has to be postponed because no doses are available. The larger the fraction of days with no stocks, the worse the situation. In Fig. 6, we see that we may not meet the second doses needs as often as 40%. Assuming longer periods of no arrivals (i.e., Lengthening the look-ahead period) acts a hedge against worse periods and strongly reduces the fraction of out-of-stock days.

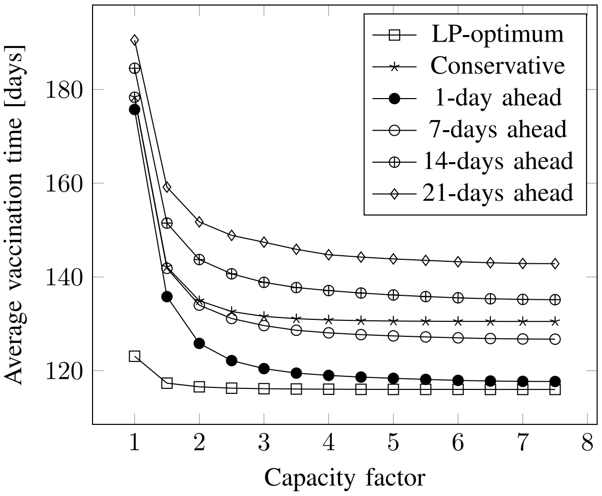


Fig. 5. Average vaccination time

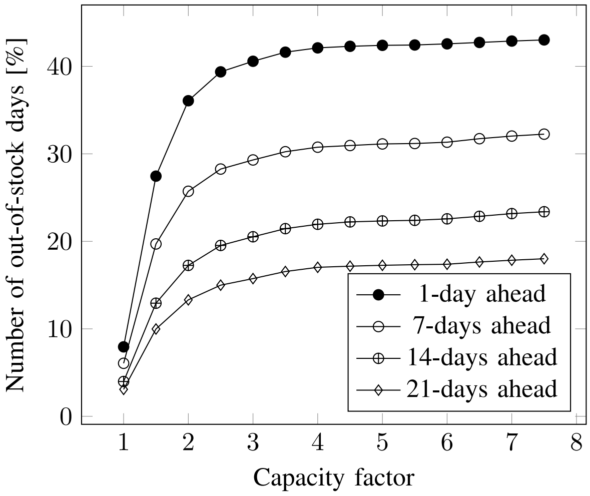


Fig. 6. Percentage of out-of-stock days

However, the relevance of out-of-stock days depends on how many doses we miss. For that reason, we also take a look at the actual backlog, which is shown in Fig. 7. The worst case takes place with the 1-day-ahead strategy, where the doses needed amount to 3% at most of the overall number of arrivals. Here, being limited by vaccination capacity helps since doses exceeding the daily capacity must be kept for use in the following days, thus acting as a reserve for days of no arrivals. Also, longer look-ahead strategies are less impacted since a longer planning period allows to override no-arrivals periods.

Since the final aim is to exploit the delivered doses as most as possible, we can analyze the utilization rate, which gives us the percentage of first-plus-second doses that have actually been administered (i.e., the percentage of vaccination cycles that have been completed) over the whole number of doses. In Fig. 8, we see that being limited by the vaccination capacity may strongly lower the capability to vaccinate as many people

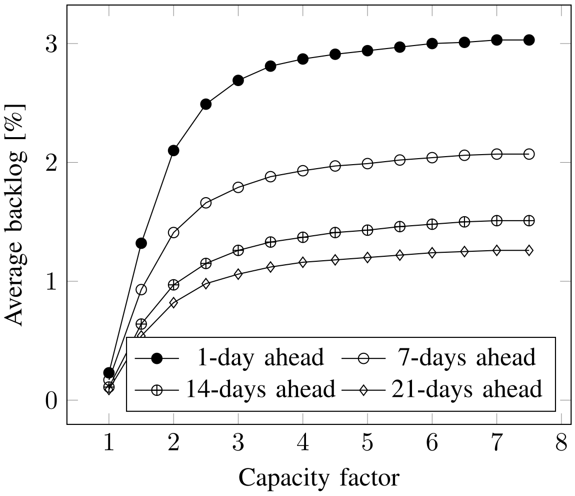


Fig. 7. Average backlog

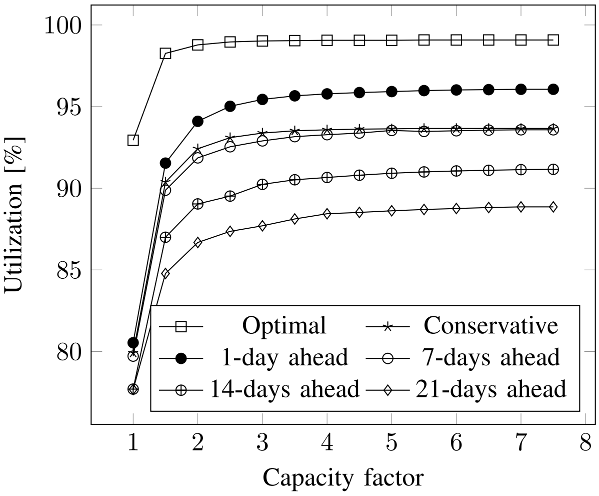


Fig. 8. Completion of vaccination cycle [%]

as possible. The curve steeply grows as we increase the capacity factor from 1 to 2. Though the clairvoyant strategy is again the best in class, as expected, with the 1-day ahead-strategy as a not-too-close runner up, the conservative strategy achieves a higher utilization than all the look-ahead strategies with a look-ahead period longer than seven days.

VII. Conclusion

In this work, we have presented a computational study to compare alternative strategies for massive vaccination under uncertain supply. The aim is to size the vaccination center capacity adequately.

The clairvoyant strategy can be set as the benchmark, leading to an optimal linear programming solution. Among the strategies examined, the best trade-off is achieved by the conservative strategy, where the administration of second doses is guaranteed on time since second doses are kept in stock as soon as the pertaining first doses are administered. Its average vaccination time is just 11% longer than what the clairvoyant strategy offers, with a utilization rate that is just 6% lower.

Among the look-ahead strategies, the 7-days-ahead strategy has very close performance regarding vaccination times and utilization, but lack of stocks is incurred in 30% of time (while this never happens with the conservative strategy). The 1-day-ahead strategy would allow reaching vaccination times just slightly longer than the clairvoyant strategy with a utilization ratio just 4% lower, but results in an unacceptable 40% fraction of out-of-stock days.

As to the sizing of the vaccination center, the capacity appears as a critical factor as long as it is too low. Capacity factors between 2 and 3 (i.e., the capability to vaccinate daily as many as 2-3 times the average number of arrivals on a day) are enough to achieve performances very close to those obtainable under infinite capacity.

As to the latter issue, we might improve the vaccination strategy by including the possibility to allow the decision-maker to change the capacities dynamically for each vaccine and each day. In practice this is not always doable, or it could be limited by a number of constraints, e.g., the severe requirements that may be imposed on the stocking devices.

Any future work will also benefit from the growing availability of data about vaccine delivery, which will allow for more accurate modelling of the process of dose arrivals. In addition, strategies envisaging mixed vaccination, i.e., adopting a different vaccine for the second dose, could incorporate knowledge about the joint distribution of arrivals for the two vaccines.

In addition, strategies that have been considered so far do not exploit any information on future supply, though uncertain, future arrivals could be categorized into a set of scenarios. In this regard, a robust optimization approach [22], [23], [24] is suitable to be devised for our mass vaccination planning problem. Several criteria have been adopted in the robust optimization literature. For instance, a widely used robustness criterion is the so-called maximin criterion, according to which the best worst-case performance has to be sought. In our case, such a robust optimization approach would maximize the system performance while guaranteeing the feasibility of the prescribed vaccine administration along the whole planning horizon for any possible future scenario.

Another important topic to be considered in future extended models for mass vaccination is the design of inducement policies to encourage the largest possible fraction of the population to uptake the vaccine. From a methodological point of view, Stackelberg game approaches (see, e.g., [25], [26]) appear as a natural direction to deal with these issues. Of course, from a different perspective, also communication strategies play a crucial role, as discussed in [27] in which the effectiveness of different health communication frames is assessed.

Finally, the reliability of the implemented decision support tool must be evaluated: methods as the one illustrated in [28] are an essential appliance to investigate the influence of any system component failure on the system functioning.

References

[1] E. J. Edwardes, *A concise history of small-pox and vaccination in Europe*. HK Lewis, 1902.

[2] D. L. Heymann and R. B. Aylward, "Mass vaccination: when and why," *Mass Vaccination: Global Aspects—Progress and Obstacles*, pp. 1–16, 2006. doi: http://dx.doi.org/10.1007/3-540-36583-4

[3] P. Homayounfar, "Process mining challenges in hospital information systems," in *2012 Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 2012, pp. 1135–1140.

[4] E. Zaitseva, V. Levashenko, and M. Rusin, "Reliability analysis of healthcare system," in *2011 Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 2011, pp. 169–175.

[5] M. Naldi, G. Nicosia, A. Pacifici, and U. Pferschy, "Profit-fairness trade-off in project selection," *Socio-Economic Planning Sciences*, vol. 67, pp. 133–146, 2019. doi: http://dx.doi.org/10.1016/j.seps.2018.10.007

[6] M. J. Keeling and P. Rohani, *Modeling Infectious Diseases in Humans and Animals*. Princeton University Press, 2011. ISBN 9781400841035. doi: http://dx.doi.org/10.1515/9781400841035

[7] E. H. Kaplan, D. L. Craft, and L. M. Wein, "Emergency response to a smallpox attack: The case for mass vaccination," *Proceedings of the National Academy of Sciences*, vol. 99, no. 16, pp. 10 935–10 940, 2002. doi: http://dx.doi.org/10.1073/pnas.162282799

[8] A. R. da Cruz, R. T. N. Cardoso, and R. H. C. Takahashi, "Multiobjective synthesis of robust vaccination policies," *Applied Soft Computing*, vol. 50, pp. 34–47, 2017. doi: http://dx.doi.org/10.1016/j.asoc.2016.11.010

[9] D. Bertsimas, J. Ivanhoe, A. Jacquillat, M. Li, A. Previero, O. S. Lami, and H. T. Bouardi, "Optimizing Vaccine Allocation to Combat the COVID-19 Pandemic," *medRxiv*, 2020. doi: http://dx.doi.org/10.1101/2020.11.17.20233213

[10] H. M. Wagner and T. M. Whitin, "Dynamic version of the economic lot size model," *Management Science*, vol. 5, pp. 89–96, 1958. doi: http://dx.doi.org/10.1287/mnsc.5.1.89

[11] W. I. Zangwill, "A backlogging model and a multi-echelon model of a dynamic economic lot size production system-a network approach," *Management Science*, vol. 15, no. 9, pp. 506–527, 1969. doi: http://dx.doi.org/10.1287/mnsc.15.9.506

[12] S. S. DeRoo, N. J. Pudalov, and L. Y. Fu, "Planning for a COVID-19 vaccination program," *Jama*, vol. 323, no. 24, pp. 2458–2459, 2020. doi: http://dx.doi.org/10.1001/jama.2020.8711

[13] R. Shretta, N. Hupert, P. Osewe, and L. J. White, "Vaccinating the world against COVID-19: getting the delivery right is the greatest challenge," *BMJ Global Health*, vol. 6, no. 3, 2021. doi: http://dx.doi.org/10.1136/bmjgh-2021-005273

[14] L. Bell and R. Wagner, "Modeling Emergency Room Arrivals Using the Poisson Process," *The College Mathematics Journal*, vol. 50, no. 5, pp. 343–350, 2019. doi: http://dx.doi.org/10.1080/07468342.2019.1662710

[15] M. Naldi, "Measurement-based modelling of internet dial-up access connections," *Computer networks*, vol. 31, no. 22, pp. 2381–2390, 1999. doi: http://dx.doi.org/10.1016/S1389-1286(99)00091-2

[16] D. K. Agarwal, A. E. Gelfand, and S. Citron-Pousty, "Zero-inflated models with application to spatial count data," *Environmental and Ecological statistics*, vol. 9, no. 4, pp. 341–355, 2002. doi: http://dx.doi.org/10.1023/A:1020910605990

[17] A. F. Zuur, E. N. Ieno, N. J. Walker, A. A. Saveliev, and G. M. Smith, *Zero-truncated and zero-inflated models for count data*. Springer, 2009, pp. 261–293. doi: http://dx.doi.org/10.1023/10.1007/978–0–387–87 458–6_11.

[18] S. Beckett, J. Jee, T. Ncube, S. Pompilus, Q. Washington, A. Singh, and N. Pal, "Zero-inflated Poisson (ZIP) distribution: Parameter estimation and applications to model data from natural calamities," *Involve, a Journal of Mathematics*, vol. 7, no. 6, pp. 751–767, 2014. doi: http://dx.doi.org/10.2140/involve.2014.7.751

[19] B. W. Silverman, *Density estimation for statistics and data analysis*. Routledge, 2018. ISBN 9780412246203 doi: http://dx.doi.org/10.1201/9781315140919

[20] S. J. Sheather and M. C. Jones, "A reliable data-based bandwidth selection method for kernel density estimation," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 53, no. 3, pp. 683–690, 1991. doi: http://dx.doi.org/0.1111/j.2517-6161.1991.tb01857.x

[21] Gurobi, "Gurobi Optimizer Reference Manual," https://www.gurobi.com/documentation/9.1/refman/index.html, 2020, [Online; accessed 24-May-2021].

[22] P. Detti, G. Nicosia, A. Pacifici, and G. Zabalo Manrique de Lara, "Robust single machine scheduling with a flexible maintenance activity," *Computers and Operations Research*, vol. 107, pp. 19–31, 2019. doi: http://dx.doi.org/10.1016/j.cor.2019.03.001

[23] A. Parnianifard, A. S. Azfanizam, M. K. A. Ariffin, and M. I. S. Ismail, "An overview on robust design hybrid metamodeling: Advanced methodology in process optimization under uncertainty," *International Journal of Industrial Engineering Computations*, vol. 9, no. 1, pp. 1–32, 2018. doi: http://dx.doi.org/10.5267/j.ijiec.2017.5.003

[24] D. S. Yamashita, V. A. Armentano, and M. Laguna, "Robust optimization models for project scheduling with resource availability cost," *Journal of Scheduling*, vol. 10, no. 1, pp. 67–76, 2007. doi: http://dx.doi.org/10.1007/s10951-006-0326-4

[25] U. Pferschy, G. Nicosia, A. Pacifici, and J. Schauer, "On the Stackelberg knapsack game," *European Journal of Operational Research*, vol. 291, no. 1, pp. 18–31, 2021. doi: http://dx.doi.org/10.1016/j.ejor.2020.09.007 Cited By 0.

[26] U. Pferschy, G. Nicosia, and A. Pacifici, "A Stackelberg knapsack game with weight control," *Theoretical Computer Science*, vol. 799, pp. 149–159, 2019. doi: http://dx.doi.org/10.1016/j.tcs.2019.10.007

[27] M. Motta, S. Sylvester, T. Callaghan, and K. Lunz-Trujillo, "Encouraging COVID-19 Vaccine Uptake Through Effective Health Communication," *Frontiers in Political Science*, vol. 3, p. 1, 2021. doi: http://dx.doi.org/10.3389/fpos.2021.630133

[28] E. Zaitseva, J. Kostolny, M. Kvassay, V. Levashenko, and K. Pancerz, "Failure analysis and estimation of the healthcare system," in *Proceedings of the 2013 Federated Conference on Computer Science and Information Systems*, M. P. M. Ganzha, L. Maciaszek, Ed. IEEE, 2013, pp. pages 235–240.

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 403–407

DOI: 10.15439/2021F32
ISSN 2300-5963 ACSIS, Vol. 25

Classification of Alzheimer's Disease Patients using Metrics of Oculo-Motors

Wioletta Nowak[*], Minoru Nakayama[†], Elzbieta Trypka[‡], Anna Zarowska[§]
[*] Institute of Biomedical Engineering and Instrumentation
Wrocław University of Science and Technology, Wrocław, Poland 50–370
Email: wioletta.nowak@pwr.edu.pl
[†] Department of Information and Communications Engineering
Tokyo Institute of Technology, Tokyo, Japan 152-8552
Email: nakayama@ict.e.titech.ac.jp
[‡] Wrocław Medical University Rektorat, Wybrzeze Ludwika Pasteura 1, 50-367 Wrocław, Poland
Email: elzbieta.trypka@umed.wroc.pl
[§] Institute of Biomedical Engineering and Instrumentation
Wrocław University of Science and Technology, Wrocław, Poland 50–370
Email: anna.zarowska@pwr.edu.pl

***Abstract*—Ocular information was observed during a set of dementia tests involving participants with Alzheimer's Disease (AD), with a mild level of cognitive impairment (MCI), or in a control group. The number of participants was 26. Features of changes in pupil size and in the central position of both eyes of participants of all three types were compared. There are significant differences in some of the metrics between the types, in earlier test sessions. The possibility of classification was confirmed using the extracted features, and the contributions of some features were examined.**

I. Introduction

HUMAN cognitive ability is often affected by disease, such as by Alzheimer's disease (AD) for example. The decline in cognitive ability of elderly people can be monitored carefully during daily exercise session [1]. In general, behavioural actions requiring some level of workload affect visual information processing and cognitive perception. The symptoms of diseases can be detected using biometrics, such as indices of oculo-motors. Changes in the level of cognitive activity and mental workload handling ability are often assessed using pupillary changes [2], [3], [4]. The symptoms of AD and aged macular disease (AMD) are also detected using pupillary reactions to the pupil light reflex (PLR) [5], [6], [7].

Most diagnostics for AD patients are based on medical consultations using cognitive tests such as the Mini-Mental State Examination (MMSE) [8] or the Montreal Cognitive Assessment (MoCA) [9]. In the test results, participants are classified into AD, mild cognitive impairment (MCI) and no cognitive impairment (NCI), or categories with normal elderly controls (NC) [9]. Since medical consultations are conducted during face-to-face oral questioning, some communication skills such as language ability and the participant's mental situation may influence the diagnostic result. Therefore, the time spent by medical practitioners and the workload of participants is not insignificant. Any appropriate procedures which can assist with medical observation using participant's behavioural metrics to reduce the difficulty of making proper assessments and permit this AD diagnostic procedure to become more widely available.

As mentioned above, ocular responses including pupillary changes and eye movement provide some evidence of symptoms of cognitive activity impairment. Observation of participant's oculo-motor changes is introduced into medical consultation, and the possibility of detecting cognitive impairment is examined. The following topics are addressed in this paper.

1) Pupillary changes and eye movement are measured using oculo-motor indices during a medical consultation to diagnose MMSE, and the metrics of the three diagnostic classes AD, MCI and NC are compared.
2) A prediction procedure for diagnostic classes is developed using the metrics measured, and its performance is discussed.

II. Experimental method

During a medical consultation of elderly people for MMSE, ocular metrics were measured.

A. Cognitive test

A cognitive exam, based on MMSE and consisting of the following 6 tests, was administered.

- Test 1: Control condition for measurement without any tasks, or baseline monitoring.
- Test 2: Orientation including simple questions and a conversation session.
- Test 3: Memory and recall tests of 5 words.
- Test 4: Oral calculation such as decrements of 7 starting from 100.
- Test 5: 2nd question and conversation session.

This work was partially supported by the Polish Ministry of Science and Higher Education research grant NN518 405338 and by the International Collaboration Research Promotion Fund, School of Engineering, Tokyo Institute of Technology.

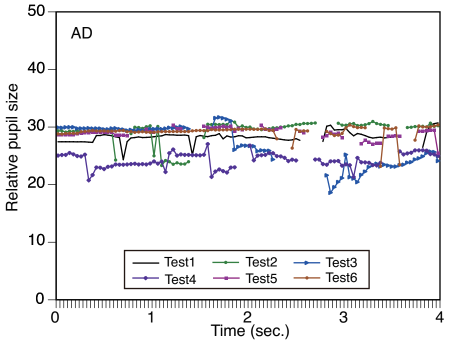


Fig. 1. Pupillary changes of an AD patient

TABLE I
MEAN OBSERVATION DURATION (SEC.) FOR PARTICIPANT GROUPS

	Test					
Sub. Group	1	2	3	4	5	6
AD (N=4)	44.6	58.4	65.7	61.1	57.1	76.4
(SD)	14.0	31.3	10.8	7.3	9.8	32.5
MCI (N=15)	33.4	62.7	53.1	39.6	36.0	42.8
(SD)	3.2	27.4	10.4	15.1	10.5	27.3
Control (N=7)	32.1	61.6	58.5	32.3	32.7	23.1
(SD)	1.1	17.5	24.4	9.5	6.3	9.1

- Test 6: Recall test for words memorised in Test 3.

The tests were conducted at a hospital in Poland and each participant was examined for AD and MCI by an M.D. who was one of the authors. Therefore, test tasks and duration varied by participant according to their medical condition.

B. Participants

In the results of the diagnoses, 26 participants (mean age: 73.7yo, STD: 9.9yo) were classified into three groups. The control group (NC) consisted of people of comparable age.

- AD: 4 (F:4, mean age: 77.3yo, STD: 6.7yo)
- MCI: 15 (F:11, M:4, mean age: 73.4yo, STD 13.1yo)
- NC: 7 (F:5, M:2, mean age: 72.4yo, STD:5.3yo)

Informed consent was obtained from all participants prior to the experiment. One MCI (F,70yo) had data partially missing, and this data was omitted from later analysis.

C. Procedure

Participants wore image tracker eye goggles (VisualEyes 505), and were asked to view the face of an M.D. during the test. Before the test, calibration of viewing points was conducted. Images of both eyes were recorded as 640×240 pixel images at 25Hz using a camera with a lens in each side of the goggles. The images were recorded using the MP4 format. After the experiment, the centre of the pupil and the size of the pupil were extracted. Fig. 1 shows the pupillary changes of an AD patient during Tests 1 through 6. The data measured during blinks was omitted in the analysis which followed. Durations where pupillary changes were dropped indicate data omitted due to blinks.

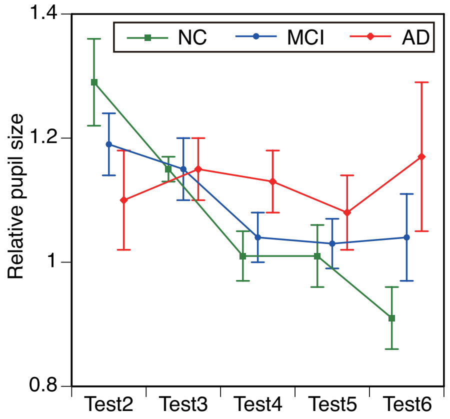


Fig. 2. Mean relative pupil sizes between groups (Error bar: STD Error)

D. Post processing

1) Pupil response: Pupil size was calculated as the sum of the pixels of each frame of the black and white video using an image processing technique. Mean pupil sizes were calculated for each duration of observation. Pupillary changes were evaluated as frequency powers of less than 4Hz, as pupils behave like low-pass filters. The frequency powers were calculated as mean power spectrum densities for 0.5∼1.6Hz and 1.6∼4.0Hz, as they were in the previous study [10]. The lowest frequency power (f<0.5Hz) was omitted due to signal noise [11].

2) Eye movement: During image processing, the centre of the pupil was measured. The measurement specification of the goggles suggests that the shift in the centre of the pupil corresponds with a visual angle of 0.25 deg for both the horizontal and vertical axes. Eye movement data was generated using this method of conversion. The eye movement was classified into fixations and saccades using a threshold of 40deg/sec [12]. For each test observation, saccade frequency, mean saccade lengths, fixation frequency, and mean fixation duration were summarised.

Since the ocular metrics of both eyes were measured, they were evaluated using a procedure which examined repeated individual measurements.

III. RESULTS

A. Observation time

As the duration of test observation depended on the condition of the individual participant, mean durations are summarised in Table I, which consists of test number and participant group. The duration of most tests except for Test 2 was around 30 seconds. While the duration varied between tests, it increased gradually as group participants progressed from NC to MCI to AD. There may have been some difficulty for participants with AD to perform some of the test tasks.

B. Pupillary changes

1) Mean pupil sizes: Pupil sizes were standardised in Test 1 using mean pupil sizes. Mean relative pupil sizes for the

TABLE II
TWO-WAY ANOVA: PARTICIPANT GROUPS × TESTS

Source	df	SS	MS	F	Sig.
Sub. Group	2	0.059	0.029	0.43	n.s.
Test	5	1.078	0.216	3.15	$p < 0.01$
Sub-G × Test	10	0.620	0.062	0.91	n.s.

TABLE III
ANOVA ANALYSIS FOR EACH SUBJECT GROUP

Sub. Group	df	SS	MS	F	Sig.
AD	5	0.154	0.031	0.35	n.s.
MCI	5	0.816	0.163	2.14	$p < 0.10$
NC	5	1.298	0.260	6.33	$p < 0.01$

three groups of participants are summarised in Fig. 2. Error bars indicate standard errors. The mean size of NC decreased gradually from Tests 2 and 6. Both task difficulty and order effect may have affected pupil sizes. However, mean pupil size of the AD group remained relatively large throughout the tests, and the deviations were smaller than the deviations of other groups. The means for MCI also decreased along with the tests, and the deviation was smaller than for the NC group.

Two-way ANOVA was conducted in order to examine the contributions of the factors of participant groups and test results. These results are summarised in Table II. The test factor is significant, but participant groups and the interaction of the two factors are not significant.

In further analysis, an F-test of the participants of each group was conducted. The results are summarised in Table II. Test factors were significant for both NC ($p < 0.01$) and MCI ($p < 0.05$). The results may reflect the pupil responses of participants of the groups.

2) Pupillary oscillation: As pupillary changes between tests and groups of participants was confirmed, frequency analysis was also applied. As mentioned in the section above, power spectrum densities (PSD) in the frequency ranges from 0.5 to 1.6Hz and from 1.6 to 4.0Hz were compared. There were no significant changes between tests or groups of participants.

A typical change was observed during Test 1. The PSD of pupillary changes of each group is summarised in Fig. 3. As the figure shows, there are some differences between groups of participants. The results of F-tests of the means of PSDs are summarised in Table IV. There is a significant difference ($p < 0.01$) between groups in the high frequency range (1.6∼4.0Hz), and the tendency for a difference to exist ($p < 0.10$) in the low frequency range (0.5∼1.6Hz).

This result suggests that differences in PSDs appear during controlled conditions, such as in Test 1.

C. Eye movement

Features of eye movement, which were tracked using the above mentioned procedure, are summarised for mean saccade frequency in Fig. 4. The error bars indicate STD Errors.

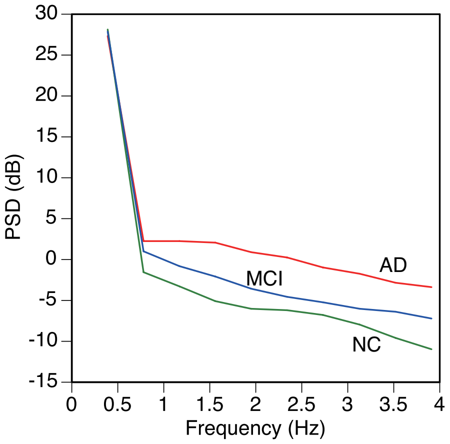


Fig. 3. PSD of pupillary change in Test 1

TABLE IV
ANOVA FOR PARTICIPANT GROUPS IN TEST 1

Frequency	df	SS	MS	F	Sig.
0.5 < f < 1.6Hz	2	6.887	3.444	2.62	$p < 0.10$
1.6 < f < 4Hz	2	2.062	1.031	5.62	$p < 0.01$

In Fig. 5, there are few differences between participant groups, but mean frequencies increase gradually during medical consultation after Test 1. The change in frequencies during complete tests was examined using two-way ANOVA, and the factor for tests was significant ($F(5, 294) = 28.4, p < 0.01$), however the factor for groups of participants was not significant ($F(2, 294) = 1.8, p = 0.16$). The effect of groups of participants on each test was examined, the while factor in Test 1 ($F(2, 49) = 7.7, p < 0.01$) was significant, there were significant differences between groups except for one pair of MCI and NC. Even in Test 2, the factor for groups of participants showed a tendency to be different ($F(2, 49) = 2.48, p < 0.10$), but overall there were no significant differences between groups.

Changes in mean saccade lengths are summarised in Fig.

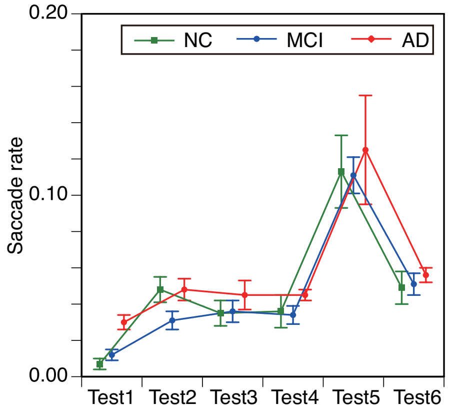


Fig. 4. Saccade frequency (deg./sec.)

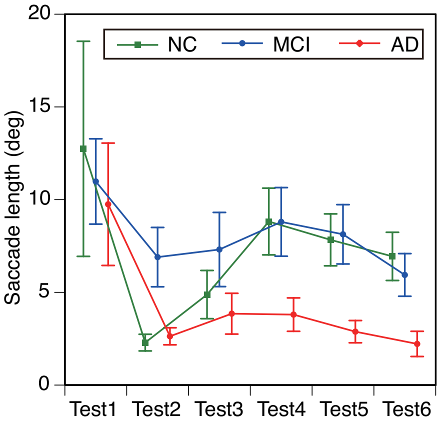


Fig. 5. Saccade length (deg.)

TABLE V
CLASSIFICATION RESULTS USING TESTS 1 AND 2

Sub Grp	(1)	(2)	(3)	Recall	F1	Pred. pairs
(1)AD	6	2	0	0.75	0.80	ad-ad:3/4
(2)MCI	0	25	3	0.89	0.85	mci-*:14/14
(3)NC	1	4	9	0.64	0.69	nc-*:5/7
Precision	0.86	0.81	0.75	P-mean:0.81; R-mean:0.76		

Rate of Lower Triangular Matrix: 0.90

5. The effect of groups of participants was only examined in Test 2 ($F(2,48) = 3.0, p < 0.10$), but there were no significant differences between groups. Also, a significant different tendency for groups of participants to be different was observed in the fixation frequency of Tests 1 and 2. Therefore, typical eye movement responses may appear in initial tests, as they did in Tests 1 and 2.

In this study, conversations during medical consultation were not analysed, though some mental workloads were detected in oculo-motor indices during Tests 2 through 6. The results suggest that there are significant differences between the metrics of Test 1 and the metrics of the other tests. However, the deviations in metrics measured during tests may be not large. Therefore, the metrics can be noted as means of duration for the tests.

D. *Prediction of groups of participants*

Since there were some significant differences in the metrics of the groups of participants extracted, the possibility of predicting the class of the group was examined using the features extracted.

1) Prediction using the observed metrics: Three groups of participants were estimated in 6 tests using 6 metrics (mean pupil size, mean PSD of pupil oscillation, mean saccade frequency, mean saccade length, mean fixation frequency, and mean fixation duration). An estimation model was trained using the Random Forest method.

Estimation was conducted using the 6-dimensional data from Tests 1 to 6, and performance was examined using trial and error. As a qualitative evaluation, performance during Tests 1 and 2 was better. The results are summarised in the contingency table shown in Table V. The vertical cells represent groups of participants, and the horizontal cells represent predictions. As the predictions were conducted using both eyes, the overall numbers are twice the number of participants. The prediction performance recall rate, precision rate and F1 indices were calculated. Also, correct prediction by either eye is summarised in the rightmost cell of Table V.

TABLE VI
CLASSIFICATION RESULTS USING TESTS 1 TO 6 WITH DATA EXTENSIONS

Sub Grp	(1)	(2)	(3)	Recall	F1	Pred. pairs
(1)AD	8	0	0	1.00	0.64	ad-ad:4/4
(2)MCI	7	20	1	0.71	0.71	mci-*:13/14
(3)NC	2	5	7	0.50	0.64	nc-*:4/7
Precision	0.47	0.71	0.88	P-mean:0.69; R-mean:0.74		

Rate of Lower Triangular Matrix: 0.98

TABLE VII
CLASSIFICATION RESULTS USING TESTS 1 AND 2 WITH DATA EXTENSIONS

Sub Grp	(1)	(2)	(3)	Recall	F1	Pred. pairs
(1)AD	8	0	0	1.00	0.64	ad-ad:4/4
(2)MCI	6	21	1	0.77	0.82	mci-*:12/14
(3)NC	3	2	9	0.64	0.75	nc-*:6/7
Precision	0.47	0.91	0.90	P-mean:0.76; R-mean:0.80		

Rate of Lower Triangular Matrix: 0.98

The results of the table suggest the possibility of predicting a participant's group using the metrics of eye observations. The purpose is a targeted diagnostic procedure which can predict the most-advanced levels of AD. Therefore, the number of predictions of the lower triangular matrix may be another index of performance. The highest recall rate appears with the MCI group, as the number of participants is the largest. The number of participants may be an influence bias regarding prediction performance.

2) Prediction with data extensions: The medical consultation data set seems insufficient because it is not easy to gather a large and balanced number of participants for each group. A data extension technique was introduced in order to obtain a set of prepared data. Here, all metrics were hypothesised to deviate using Gaussian distribution $N(\mu, \sigma)$, and the range of re-sampled data was controlled between $\mu \pm k\sigma (k = 1, 1.5, 2)$. As a result, 300 sets of data were prepared in total, with 100 sets of data being generated for each participant group.

The prediction models were trained with the data generated using the Random Forest technique once again, and actual measured data was tested. Recall performance between k parameters was compared, and $k = 1.5$ produced the highest level of performance.

The test results are presented in Table VI using the same format as the previous table, where all generated data for Tests 1 to 6 is employed as the set of trained data. The results show that for all participants in the AD group, AD was able to be predicted using both eyes. The data extensions may have compensated for the unbalanced data samples. However, incorrect predictions for MCI and NC groups increased, and

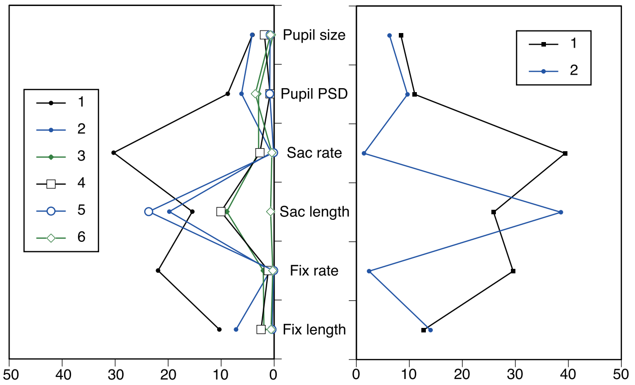


Fig. 6. Comparison of Gini coefficients of two conditions. Left side: Tests 1∼6, Right side: Tests 1∼2

some indices of performance decreased slightly. The number of predictions of the lower triangular matrix was sufficiently high, and thus the possibility of using prediction as a diagnostic procedure was confirmed.

In the previous section, features of metrics of patients in the early stages of consultation were suitable for prediction. A second set of results is summarised in Table VII, where features of Tests 1 and 2 are limited. In these results, NC group prediction performance improved, and some of the performance metrics showed the highest rates of correct prediction.

For a detailed analysis of performance, the contributions of features of observed metrics between the two models are compared in Tables VI and VII. The degree of contribution of prediction as an index of importance (the Gini coefficient) is summarised in Fig. 6. The left side represents coefficients in Tests 1 to 6, and the right side represents coefficients in Tests 1 to 2. The figure suggests that the coefficients for Test 1, in particular saccade frequency, saccade length and fixation frequency, present a larger contribution than do the features of pupil responses. The coefficients on the left side suggest that most coefficient in Tests 3 to 6 were relatively small. These results confirm the importance of using metrics in the early stages of the consultation, as the changes between the initial cognitive load and the control session may represent the situation of individual participants.

The results show that there were differences in measured metrics between groups of participants, and that significant differences were observed during the early stages of testing, including during the control session which was the first stage of the consultation. The prediction procedure was conducted using the Random Forest and data extension techniques. The accuracy of the prediction procedure was confirmed.

IV. Summary

In this paper, changes in pupil responses and eye movement between AD, MCI and NC participants during cognitive tests were compared. The tests consisted of using MMSE as a form of medical consultation. Also, the possibility of estimating participant groups was examined.

Validation of the procedure and improvement of prediction performance will be subjects of our further study.

References

[1] K. Aoki, T. T. Ngo, I. Mitsugami, F. Okura, M. Niwa, Y. Makihara, Y. Yagi, and H. Kazui, "Early detection of lower MMSE scores in elderly based on dual-task gait," *IEEE Access*, vol. 7, pp. 40 085–40 094, 2019.

[2] J. Beatty, "Task-evoked pupillary response, processing load, and the structure of processing resources," *Psychological Bulletin*, vol. 91, no. 2, pp. 276–292, 1982.

[3] J. Kuhlmann and M. Böttcher, Eds., *Pupillography: Principles, Methods and Applications*. Munchen, Germany: W. Zuckschwerdt Verlag, 1999.

[4] M. Nakayama and M. Katsukura, "Development of a system usability assessment procedure using oculo-motors for input operation," *Universal Access in Information Society*, vol. 10, no. 1, pp. 51–68, 2011.

[5] M. Nakayama, W. Nowak, H. Ishikawa, K. Asakawa, and Y. Ichibe, "Discovering irregular pupil light responses to chromatic stimuli using waveform shapes of pupillograms," *EURASIP J. in Bioinformatics and System Biology*, vol. #18, pp. 1–14, 2014.

[6] W. Nowak, M. Nakayama, T. Kręcicki, E. Trypka, A. Andrzejak, and A. Hachoł, "Analysis for extracted features of pupil light reflex to chromatic stimuli in Alzheimer's patients," *EAI Endorsed Transactions on Pervasive Health and Technology*, vol. 5, pp. 1–10, November 2019, e4.

[7] W. Nowak, M. Nakayama, T. Kręcicki, and A. Hachoł, "Detection procedures for patients of Alzheimer's disease using waveform features of pupil light reflex in response to chromatic stimuli," *EAI Endorsed Transactions on Pervasive Health and Technology*, vol. 6, pp. 1–11, December 2020, e6.

[8] M. F. Folstein, S. E. Folstein, and P. R. McHugh, "MINI-MENTAL STATE – a practical method for grading the cognitive state of patients for the clinician," *Journal of Psychiatric Research*, vol. 12, pp. 189–198, 1975.

[9] Z. S. Nasreddine, N. A. Phillips, V. Bédirian, S. Charbonneau, V. Whitehead, I. Collin, J. L. Cummings, and H. Chertkow, "The Montreal Cognitive Assessment, MoCA: A brief screening tool for mild cognitive impairment," *Journal of American Geriatrics Society*, vol. 53, pp. 695–699, 2005.

[10] V. Peysakhovich, M. Causse, S. Scannella, and F. Dehais, "Frequency analysis of a task-evoked pupillary response: Luminance-independent measure of mental effort," *International Journal of Psychophysiology*, vol. 97, pp. 30–37, 2015.

[11] K. Ukai, "Pupil," in *SHIKAKU JYOUHOU SYORI HANDOBUKKU (Handbook of Visual Information Processing)*, Japan Society of Vision Science, Ed. Tokyo, Japan: Asakura Shoten, 2001.

[12] Y. Ebisawa and M. Sugiura, "Influences of target and fixation point conditions on characteristics of visually guided voluntary saccade," *The Journal of the Institute of Image Information and Television Engineers*, vol. 52, no. 11, pp. 1730–1737, 1998.

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 409–413
DOI: 10.15439/2021F40
ISSN 2300-5963 ACSIS, Vol. 25

Smartphone-Based Color Measurement of Tooth Shade Guide in Clinical Lighting Conditions

Agnieszka Radziun,
Rafał Doniec, Szymon Sieciński,
Natalia Piaseczna, Konrad Duraj,
Ewaryst Tkacz
Silesian University of Technology
Faculty of Biomedical Engineering
Department of Biosensors
and Processing of Biomedical Signals
F. D. Roosevelta 40,
41-800 Zabrze, Poland
Email: agnieszkaaradziun@gmail.com,
rafal.doniec@polsl.pl
szymon.siecinski@polsl.pl
natalia.piaseczna@polsl.pl
kondrad.duraj@polsl.pl, etkacz@polsl.pl

Katarzyna Mocny-Pachońska,
Marta Tanasiewicz
Department of Conservative
Dentistry with Endodontics,
Faculty of Medical Sciences in Zabrze,
Medical University of Silesia
Plac Akademicki 17
41-902 Bytom, Poland
Email: kpachonska@sum.edu.pl
martatanasiewicz@sum.edu.pl

Marta Cieślik-Wegemund
Department of Periodontal
and Oral Mucosa Diseases,
Faculty of Medical Sciences in Zabrze,
Medical University of Silesia,
Plac Traugutta 2
41-800 Zabrze, Poland
Email: martawegemund@gmail.com

***Abstract*—Caries is a common disease of hard tissues of teeth which results in dental cavities, which are usually replaced by dental fillings. Matching the color of a dental filling is usually a subjective assessment. In this study, we conducted a color analysis of GC Gradia Direct shade guide in the lighting conditions of the dental office. Color measurement was performed using Color Grab mobile app and the results were acquired as values of RGB (red, green, blue) and HSV (hue, saturation, color value) values. The results indicate the possibility of identifying each shade of tooth by the most prominent changes in RGB and/or HSV components.**

I. Introduction

CARIES is a common disease of hard tissues of the teeth caused by bacteria, which leads to the demineralization and proteolytic decay [1], [2]. The damage of tooth hard tissue associated with caries is defined as cavity [3]. Due to the fact that the damaged structure of the tooth does not regenerate, it must be replaced by a filling. Restorative materials include gold, dental amalgam, composite, and porcelain resins [4].

One of the composite materials used for fillings to match the color to the patient's teeth is GC Gradia Direct dental composite material (GC Europe NV, Leuven, Belgium), which recreates the optical properties of natural teeth and is available in several versions that differ in color scale [5].

Color as an optical value significantly affects the appearance of the tooth [6], [7]. The color depends not only on the shade, but also on the saturation, brightness, and tooth morphology [7]. The perception of tooth color depends on the way it reflects the light [6], [8].

In most cases, a dentist matches the filling color by the subjective assessment based on comparison of the shade of tooth with the shade guide [6], [7]. To reduce the effects of metamerism, a standardized light source should be used [6]. Three most common light sources in dentistry are natural, fluorescent, and incandescent [9]. Measurement of color of a wide range of materials and substrates has been performed by spectrophotometers and colorimeters for several years [10], also in dentistry [11], [12].

Spectrophotometers measure the amount of light absorbed by the sample using the Lambert Beer law. They consist of their own light source, monochromator, a sample, and photodetector [10]. Due to the technological progress and the decreasing price of devices, the use of spectrophotometers in dentistry has become a viable alternative to visual matching [13], [14]. Another method for tooth color analysis is using dental microscopes [2].

Due to the increasing capabilities of smartphones, there have been proposed several applications, including color measurements [15], [16]. For instance, Kim et al. [15] applied smartphones to perform colorimetric pH measurement. Hasan et al. proposed a method for measuring hemoglobin level with smartphones [16].

The purpose of the study was to perform the color analysis of a commercially available shade guide (GC Gradia Direct) with mobile apps in the lighting conditions of the dental office and analyze the influence of light intensity and energy (visible light or ultraviolet light) on the color components of available shades in RGB and HSV color spaces, which may be useful in distinguishing the available shades.

This work was not supported by any organization.

II. Material and Methods

A. Material

Our study involved only a commercially available tooth shade guide (GC Gradia Direct produced by GC Europe NV, Leuven, Belgium) which contains the shades of teeth in the form of a palette with wedge-shaped color samples (see Fig. 1). The order of the colors in the shade guide is as follows: BW, A1, A2, A3, A3.5, A4, B2, B3, C3, CV, CVD, WT, DT, CT, NT and GT [5, pp. 6–7].

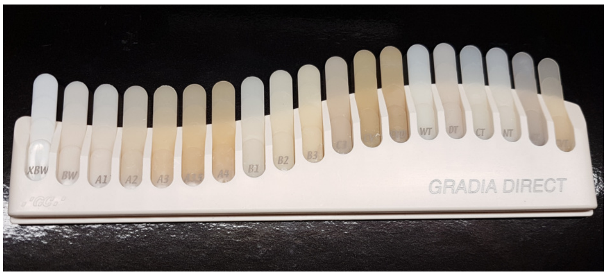


Fig. 1. GC Gradia Direct shade guide (own source).

B. Experiments

The study was conducted on a commercially available tooth shade guide (GC Gradia Direct) in typical lighting conditions of a dental office (visible light and ultraviolet (UV) light additionally lit by daylight through the windows). We used a smartphone with a camera, its own light source, and two mobile apps for Android: Lux Light Meter (Marcel Waldau Webdesign) [17] and Color Grab (Loomatix) available at the Google Play Store. Color Grab app performs real-time color callibration [18], whereas Lux Light Meter has its own callibration mechanism [17].

The first app allowed the measurement of light intensity E in lux (lx) at the test site, and the second app measured RGB (red, green, blue) and HSV (hue, saturation, color intensity value) component values in the shade guide. RGB and HSV values were taken under visible light for the following illuminance values: 130 lx, 165 lx, and 201 lx, and also under the LE-900 ultraviolet lamp with the power input of 90 W; output voltage and current of 12 V DC, 1 A; input voltage of 100-240 V AC, 50/60 Hz, and illuminance of 50 lx.

RGB values were expressed in the range of 0–255. Hue values were expressed in the range of 0–360°, saturation, and color intensity values were expressed in percents (0-100%). RGB values were converted to HSV values based on the transformation algorithm by Smith in [19].

C. Analysis of results

The obtained color components in RGB and HSV color spaces were analyzed visually to indicate the most prominent components. Then, the differences between the shades coded as shown in table III were further analyzed quantitatively by calculating the percentage of unique values of components, Spearman's rank correlation, and Pearson's linear correlation coefficient for the shades in UV and visible light to evaluate the monotonicity and linearity of the changes in color components between the shades which were assigned to an ordinal scale [20], [21].

For the shades in visible light, the Kruskal-Wallis test was additionally conducted to evaluate the significance of changes of the most prominent components in different light intensities. The quantitative analyses were conducted with MATLAB R2020b (MathWorks, Inc., Natick, MA, USA) and Pandas Profiling version 2.11.0 running under Python 3.9.

III. Results

During the study, not all locations in the dental office had the same light intensity because two types of lamps were used and some places were additionally lit by sunlight due to the location of the window. The results of color measurements under visible light are presented in subsection III-A and the results of color measurement under ultraviolet light are presented in subsection III-B.

A. Visible light

The RGB and HSV values of shades in the GC Gradia Direct shade guide measured under visible light are shown in Tab. I and Tab. II.

Each shade in the shade guide has its own value. RGB values are significantly influenced by the changes of light intensity, as observed in table I.

The HSV values of the same shades in the GC Gradia Direct in visible light (see table II) have very similar color intensity value for $E = 130$ lx and $E = 165$ lx. For $E = 201$ lx, the V values of the analyzed shades are more significant and may help distinguish each shade. More prominent feature of the shades are hue and saturation for all analyzed light intensities in visible light.

The relationships of values of color components with the shades for all considered illuminances were also evaluated quantitatively by calculating the Spearman's rank correlation and Pearson's linear correlation coefficients (see table IV), and Kruskal-Wallis test after assigning numbers from 0 to 15 for the shades (see table III). The percentage of distinct component values are shown in table V.

The r values shown in Tab. IV show that the relationship between color components (blue in RGB, hue and saturation in HSV color spaces) and tooth shade is not monotonic and/or linear, except for the hue component in 130 lx ($r = 0.556$, $p < 0.05$), which is monotonic. However, the differences between the blue component, hue, and saturation for each shade as coded in Tab. III in visible light are significant in Kruskal-Wallis test for $p < 0.05$ (see figures 2, 3, and 4 and tables VI, VII, and VIII). In Kruskal-Wallis test results, SS is the sum of squares due to each source, df is the number of degrees of freedom associated with each source, MS are the mean squares for each source, and χ^2 is the test statistic.

TABLE I
RGB VALUES OF GC GRADIA DIRECT SHADES IN VISIBLE LIGHT.

E	Shade	BW	A1	A2	A3	A3.5	A4	B2	B3	C3	CV	CVD	WT	DT	CT	NT	GT
130 lx	Red	255	254	253	253	254	254	254	254	251	253	253	254	254	252	253	240
	Green	255	253	252	247	248	246	254	254	249	248	248	254	254	254	254	241
	Blue	250	241	237	227	225	220	243	238	233	226	224	252	247	248	254	233
165 lx	Red	254	254	250	254	255	254	253	252	245	248	244	254	252	250	251	242
	Green	254	254	253	250	252	246	252	248	239	241	237	254	254	251	252	244
	Blue	250	249	244	226	230	217	235	220	218	212	212	242	252	242	247	240
201 lx	Red	254	248	245	243	211	222	229	230	254	251	243	233	241	249	250	228
	Green	254	252	248	233	206	208	230	219	253	250	209	240	241	250	251	236
	Blue	246	239	225	200	174	168	203	180	235	230	209	225	220	234	236	229

TABLE II
HSV VALUES OF GC GRADIA DIRECT SHADES IN VISIBLE LIGHT

E	Shade	BW	A1	A2	A3	A3.5	A4	B2	B3	C3	CV	CVD	WT	DT	CT	NT	GT
130 lx	H [°]	60	55	56	46	48	46	60	60	53	49	50	60	60	80	180	67
	S [%]	2	5	6	10	11	13	4	6	7	11	11	1	3	2	0	3
	V [%]	100	99.6	99.2	99.21	99.6	99.6	99.6	99.6	98.43	99.21	99.21	99.6	99.6	99.6	99.6	94.5
165 lx	H [°]	60	60	120.7	51.42	52.8	47.02	56.67	52.5	46.67	48.33	46.87	60	120.4	120.2	120.2	120.4
	S [%]	1.57	1.96	3.55	11.02	9.8	14.56	7.11	12.69	11.02	14.51	13.11	4.72	0.78	3.58	1.98	1.63
	V [%]	99.6	99.6	99.21	99.6	100	99.6	99.21	98.82	96.07	97.25	95.68	99.6	99.6	98.43	98.82	95.68
201 lx	H [°]	60	78	68	46	52	44	62	47	57	57	0	88	60	63.75	64	127.5
	S [%]	3	5	9	18	17.53	24.32	11.73	21.73	7.48	8.36	13.99	6.25	8.71	6.4	5.97	3.38
	V [%]	99.6	98.82	97.25	95.29	82.74	87.05	90.19	90.19	99.6	98.43	95.29	94.11	94.50	98.03	98.43	92.54

TABLE III
CODE ASSIGNMENTS FOR AVAILABLE TOOTH SHADES.

Shade	BW	A1	A2	A3	A3.5	A4	B2	B3	C3	CV	CVD	WT	DT	CT	NT	GT
Code	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

TABLE IV
SPEARMAN'S RANK CORRELATION AND PEARSON'S LINEAR CORRELATION BETWEEN THE SHADES IN VISIBLE LIGHT.

Spearman's rank correlation						
Illuminance	130 lx		165 lx		201 lx	
Component	r	p	r	p	r	p
Blue	0.219	0.415	-0.049	0.858	0.196	0.466
Hue	0.556	0.025	0.218	0.417	0.253	0.344
Saturation	-0.380	0.147	-0.100	0.713	-0.265	0.321
Pearson's linear correlation						
Blue	0.235	0.382	-0.012	0.965	0.090	0.741
Hue	0.478	0.062	0.499	0.049	0.252	0.347
Saturation	-0.364	-0.153	-0.153	0.713	-0.227	0.398

TABLE V
PERCENTAGES OF UNIQUE VALUES OF COLOR COMPONENTS.

Visible light			
Illuminance	130 lx	165 lx	201 lx
Component	Uniqueness [%]	Uniqueness [%]	Uniqueness [%]
Blue	93.8	87.5	93.8
Hue	68.8	87.5	87.5
Saturation	68.8	100	100
Ultraviolet light			
Illuminance	50 lx		
Component	Uniqueness [%]		
Green	100		
Hue	87.5		

TABLE VI
RESULTS OF KRUSKAL-WALLIS TEST FOR BLUE.

Source	SS	df	MS	χ^2	p
Columns	5600	15	373.333	28.6	0.0181
Error	3603.5	32	112.609		
Total	9203.5	47			

B. UV light

Because the changes of the measured RGB components of the shades were the most prominent for B component, the color measurement was retaken under ultraviolet light. The RGB and HSV values are shown in table IX.

In RGB color space, the most prominent features of the analyzed shades are green component values, whereas the most prominent features of the shades in HSV color model are hue, for shades BW, A1 and NT, and saturation with the color intensity value for A4 shade. These observations were

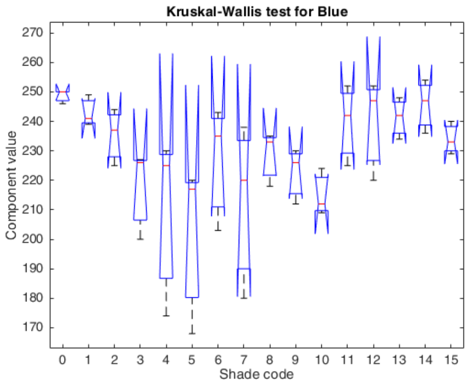


Fig. 2. The boxplot for blue component between the analyzed shades.

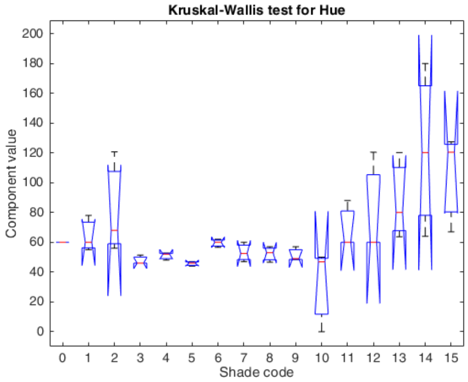


Fig. 3. The boxplot for hue between the analyzed shades.

TABLE VII
RESULTS OF KRUSKAL-WALLIS TEST FOR HUE.

Source	SS	df	MS	χ^2	p
Columns	7431.17	15	495.411	38.27	0.008
Error	1695.83	32	52.995		
Total	9127	47			

TABLE VIII
RESULTS OF KRUSKAL-WALLIS TEST FOR SATURATION.

Source	SS	df	MS	χ^2	p
Columns	6971	15	464.733	35.59	0.0002
Error	2235.5	32	69.859		
Total	9206.5	47			

confirmed by calculating the uniqueness of component values expressed as the percentage of the unique values in table V.

The r values shown in table X show that the association between analyzed variables (green in RGB, hue in HSV, color spaces, and tooth shades) is not monotonic and/or linear for $p < 0.05$.

IV. DISCUSSION

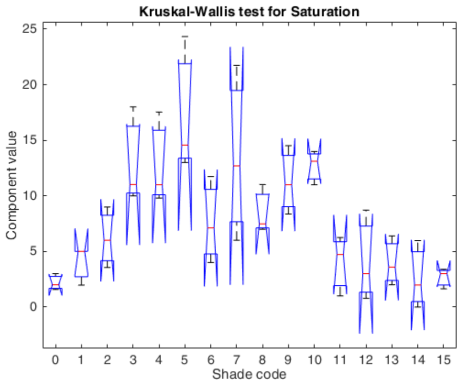


Fig. 4. The boxplot for saturation between the analyzed shades.

We measured the colors of the GC Gradia Shade Guide in a dental office using a smartphone. Measuring color changes using a smartphone is a challenging task due to the light conditions and non-linear relationship between the light intensity and the RGB of the shade [22]. Aforementioned statement also applies to HSV and CMYK color spaces. These observations were confirmed quantitatively by the Spearman's rank correlation and Pearson's linear correlation coefficients presented in tables IV and X.

The values of components in RGB, HSV and CMYK color spaces depend on the power of the light source, the influence of the daylight, position of the light source, and the distance between the camera and the measured object.

The most prominent features of shades under visible light are the value of blue component in RGB color space, hue, and saturation in HSV color space. These findings were also supported by calculating the uniqueness of values of color components and the results of Kruskal-Wallis tests.

The most prominent feature in ultraviolet light was the green component in RGB model, and hue component in HSV model. That finding was confirmed by calculating the uniqueness of color component values shown in table V. That means that each shade of tooth may be identified by the most prominent changes in RGB and/or HSV components, namely blue, hue, and saturation in visible light, and green and hue in ultraviolet light.

The findings of our study may help develop a model of teeth shades to improve dental care by optimizing the process of color matching in the preparation of dental fillings despite using other approaches to spectrophotometry than described in the literature [11], [12].

The most significant limitation of our study is no evaluation of the influence of daylight on the measured color components and the fact that using smartphone apps in illuminance measurement is not recommended due to the discepancies in measured values between the devices and various apps [23], [24]. However, smartphone cameras are suitable for color measurements [15], [16].

In future studies, we consider other shade guides, more light intensities, and minimizing the influence of daylight and non-uniform light distribution on color measurement. We

TABLE IX
RGB AND HSV VALUES OF GC GRADIA DIRECT SHADES IN UV LIGHT AND E = 50LX.

Shade	BW	A1	A2	A3	A3.5	A4	B2	B3	C3	CV	CVD	WT	DT	CT	NT	GT
Red	12	36	0	0	0	0	0	0	0	0	0	0	0	0	7	0
Green	214	204	175	133	105	66	189	179	120	56	25	13	185	208	199	128
Blue	255	255	255	255	255	255	255	255	255	255	255	217	255	255	255	255
H [°]	190	194	199	209	215	224	196	198	212	227	234	236	196	191	194	210
S [%]	95.29	85.88	100	100	100	100	100	100	100	100	100	100	100	100	97.25	100
V [%]	100	100	100	100	100	100	100	100	100	100	100	85,09	100	100	100	100

TABLE X
SPEARMAN'S RANK CORRELATION AND PEARSON'S LINEAR CORRELATION BETWEEN THE SHADES IN ULTRAVIOLET LIGHT.

	Spearman's		Pearson's	
Component	r	p	r	p
Green	-0.182	0.415	-0.190	0.482
Hue	-0.155	0.567	-0.180	0.505
Magenta	0.182	0.498	0.189	0.484

also consider the development of a mobile app for tooth color measurements which may be used to lower the cost of matching optimal shades of teeth in comparison with buying a commecially available dental spectrophotometer, especially in whitening the color of teeth or filling the cavities with dental fillings.

REFERENCES

[1] H. Silk, "Diseases of the mouth," *Primary Care: Clinics in Office Practice*, vol. 41, no. 1, pp. 75 – 90, 2014. doi: 10.1016/j.pop.2013.10.011 Primary Care ENT.

[2] O. Osadcha, A. Trzcionka, K. Pachońska, and M. Pachoński, "Detection of dental filling using pixels color recognition," in *Information and Software Technologies*, R. Damaševičius and G. Vasiljevienė, Eds. Cham: Springer International Publishing, 2018. doi: 10.1007/978-3-319-99972-2_28. ISBN 978-3-319-99972-2 pp. 347–356.

[3] L. Bjørndal, S. Simon, P. L. Tomson, and H. F. Duncan, "Management of deep caries and the exposed pulp," *International Endodontic Journal*, vol. 52, no. 7, pp. 949–973, 2019. doi: https://doi.org/10.1111/iej.13128

[4] D. Bratthall, P. E. Petersen, J. R. Stjernswärd, and L. J. Brown, "Oral and craniofacial diseases and disorders," in *Disease Control Priorities in Developing Countries*, 2nd ed., D. T. Jamison, J. G. Joel G Breman, A. R. Measham, G. Alleyne, M. Claeson, D. B. Evans, P. Jha, A. M. Mills, and P. Musgrove, Eds. World Bank, Washington (DC), 2006, ch. 38, pp. 723–736. ISBN 0821361791. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK11725/

[5] "GC Gradia Direct light-cured composite restorative clinical guide," accessed 21 March 2021. [Online]. Available: https://cdn.gceurope.com/v1/PID/gradiadirect/manual/MAN_Gradia_Direct_Clinical_Guide_en.pdf

[6] A. Watts and M. Addy, "Tooth discolouration and staining: a review of the literature," *British Dental Journal*, vol. 190, no. 6, pp. 309–316, March 2001. doi: https://dx.doi.org/10.1038/sj.bdj.4800959

[7] A. Joiner, "Tooth colour: a review of the literature," *Journal of Dentistry*, vol. 32, pp. 3–12, 2004. doi: https://doi.org/10.1016/j.jdent.2003.10.013 Advances in tooth whitening: A new, simple and effective solution.

[8] K. W. Aschheim and B. A. Singer, "Fundamentals of esthetics and smile analysis," in *Esthetic Dentistry*. Elsevier, 2015, pp. 38–54.

[9] L. Miller, "Organizing color in dentistry," *The Journal of the American Dental Association*, vol. 115, pp. 26E–40E, Dec. 1987. doi: https://dx.doi.org/10.14219/jada.archive.1987.0315

[10] T. A. Germer, J. C. Zwinkels, and B. K. Tsai, "Chapter 1 - introduction," in *Spectrophotometry*, ser. Experimental Methods in the Physical Sciences, T. A. Germer, J. C. Zwinkels, and B. K. Tsai, Eds. Academic Press, 2014, vol. 46, pp. 1–9.

[11] C. Igiel, M. Weyrauch, S. Wentaschek, H. Scheller, and K. M. Lehmann, "Dental color matching: A comparison between visual and instrumental methods," *Dental Materials Journal*, vol. 35, no. 1, pp. 63–69, 2016. doi: https://dx.doi.org/10.4012/dmj.2015-006

[12] J. C. Ragain, "A review of color science in dentistry: Colorimetry and color space," *Journal of Dentistry, Oral Disorders & Therapy*, vol. 4, no. 1, pp. 01–05, Jan. 2016. [Online]. Available: https://doi.org/10.15226/jdodt.2016.00148

[13] J.-Y. Chang, W.-C. Chen, T.-K. Huang, J.-C. Wang, P.-S. Fu, J.-H. Chen, and C.-C. Hung, "Evaluation of the accuracy and limitations of three tooth-color measuring machines," *Journal of Dental Sciences*, vol. 10, no. 1, pp. 16–20, Mar. 2015. doi: https://dx.doi.org/10.1016/j.jds.2013.04.004

[14] M. H. Kalantari, S. A. Ghoraishian, and M. Mohaghegh, "Evaluation of accuracy of shade selection using two spectrophotometer systems: Vita Easyshade and Degudent Shadepilot," *European Journal of Dentistry*, vol. 11, no. 02, pp. 196–200, Apr. 2017. doi: https://dx.doi.org/10.4103/ejd.ejd_195_16

[15] S. D. Kim, Y. Koo, and Y. Yun, "A smartphone-based automatic measurement method for colorimetric pH detection using a color adaptation algorithm," *Sensors*, vol. 17, no. 7, p. 1604, Jul 2017. doi: http://dx.doi.org/10.3390/s17071604

[16] M. K. Hasan, M. Haque, N. Sakib, R. Love, and S. I. Ahamed, "Smartphone-based human hemoglobin level measurement analyzing pixel intensity of a fingertip video on different color spaces," *Smart Health*, vol. 5-6, pp. 26–39, Jan. 2018. doi: https://doi.org/10.1016/j.smhl.2017.11.003

[17] Waldau Webdesign, "Lux light meter," Google Play Store: https://play.google.com/store/apps/details?id=de.waldau_webdesign.lightmeter, 2019, accessed 22 May 2021.

[18] Loomatix, "Color grab," Google Play Store: https://play.google.com/store/apps/details?id=com.loomatix.colorgrab, 2019, accessed 22 May 2021.

[19] A. R. Smith, "Color gamut transform pairs," in *Proceedings of the 5th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '78. New York, NY, USA: Association for Computing Machinery, 1978. doi: https://dx.doi.org/10.1145/800248.807361. ISBN 9781450379083 p. 12–19.

[20] C. Spearman, "The proof and measurement of association between two things," *The American Journal of Psychology*, vol. 15, no. 1, p. 72, Jan. 1904. [Online]. Available: https://doi.org/10.2307/1412159

[21] W. H. Kruskal and W. A. Wallis, "Use of ranks in one-criterion variance analysis," *Journal of the American Statistical Association*, vol. 47, no. 260, pp. 583–621, Dec. 1952. [Online]. Available: https://doi.org/10.1080/01621459.1952.10483441

[22] E. P. J. Tozer, *Broadcast Engineer's Reference Book*. Elsevier, 2004. ISBN 0-240-51908-6

[23] DIAL GmbH, "Luxmeter app versus measuring device: Are smartphones suitable for measuring illuminance?" Jun. 2006. [Online]. Available: https://www.dialux.com/en-GB/news-detail/luxmeter-app-versus-measuring-device-are-smartphones-suitable-for-measuring-illuminance

[24] R. Melo, F. Carvalho, and D. Cerqueira, "Pitfalls of measuring illuminance with smartphones," in *Occupational Safety and Hygiene VI*. CRC Press, Mar. 2018, pp. 459–463.

16th Conference on Information Systems Management

THIS event constitutes a forum for the exchange of ideas for practitioners and theorists working in the broad area of information systems management in organizations. The conference invites papers coming from three complimentary directions: management of information systems in an organization, uses of information systems to empower managers, and information ssytems for sutainable development. The conference is interested in all aspects of planning, organizing, resourcing, coordinating, controlling and leading the management function to ensure a smooth operation of information systems in organizations. Moreover, the papers that discuss the uses of information systems and information technology to automate or otherwise facilitate the management function are specifically welcome. Papers about the influence of information systems on sustainability are also expected.

TOPICS

- Management of Information Systems in an Organization:
  - Modern IT project management methods
  - User-oriented project management methods
  - Business Process Management in project management
  - Managing global systems
  - Influence of Enterprise Architecture on management
  - Effectiveness of information systems
  - Efficiency of information systems
  - Security of information systems
  - Privacy consideration of information systems
  - Mobile digital platforms for information systems management
  - Cloud computing for information systems management
- Uses of Information Systems to Empower Managers
  - Achieving alignment of business and information technology
  - Assessing business value of information systems
  - Risk factors in information systems projects
  - IT governance
  - Sourcing, selecting and delivering information systems
  - Planning and organizing information systems
  - Staffing information systems
  - Coordinating information systems
  - Controlling and monitoring information systems
  - Formation of business policies for information systems
  - Portfolio management,
  - CIO and information systems management roles
- Information Systems for Sustainability
  - Sustainable business models, financial sustainability, sustainable marketing
  - Qualitative and quantitative approaches to digital sustainability
  - Decision support methods for sustainable management

TECHNICAL SESSION CHAIRS

- **Arogyaswami, Bernard,** Le Moyne University, USA
- **Chmielarz, Witold,** University of Warsaw, Poland
- **Jankowski, Jarosław,** West Pomeranian University of Technology in Szczecin, Poland
- **Karagiannis, Dimitris,** University of Vienna, Austria
- **Kisielnicki, Jerzy,** University of Warsaw, Poland
- **Ziemba, Ewa,** University of Economics in Katowice, Poland

PROGRAM COMMITTEE

- **Janis Bicevskis,** University of Latvia, Latvia
- **Alberto Cano,** Virginia Commonwealth University, United States
- **Vincenza Carchiolo,** Dipartimento di Matematica e Informatica - Universita di Catania, Italy
- **Beata Czarnacka-Chrobot,** Warsaw School of Economics, Poland
- **Pankaj Deshwal,** Netaji Subhas Institute of Technology, India
- **Robertas Damasevicius,** Silesian University of Technology, Poland
- **Monika Eisenbardt,** Univeristy of Economics Katowice, Poland
- **Marcelo Fantinato,** University of São Paulo, Brazil
- **Renata Gabryelczyk,** University of Warsaw, Poland
- **Nitza Geri,** The Open University of Israel, Israel
- **Dariusz Grabara,** University, Economics in Katowice, Poland
- **Jarosław Jankowski,** West Pomeranian University of Technology in Szczecin, Poland
- **Andrzej Kobylinski,** Warsaw School of Economics, Poland
- **Christian Leyh,** Technische Universität Dresden, Germany

- **Karolina Muszyńska,** University of Szczecin, Poland
- **Tomasz Parys,** University of Warsaw, Poland
- **Uldis Rozevskis,** University of Latvia, Latvia
- **Nina Rizun,** Gdansk University of Technology, Poland
- **Andrzej Sobczak,** Warsaw School of Economics, Poland
- **Jakub Swacha,** University of Szczecin, Poland
- **Symeon Symeonidis,** Democritus Univesity of Thrace, Greece
- **Oskar Szumski,** University of Warsaw, Poland
- **Jarosław Wątróbski,** University of Szczecin, Poland
- **Janusz Wielki,** Opole University of Technology, Poland
- **Dmitry Zaitsev,** Odessa State Environmental University, Ukraine
- **Marek Zborowski,** University of Warsaw, Poland

DOI: 10.15439/2021F61
ISSN 2300-5963 ACSIS, Vol. 25

Towards Objectification of Multi-Criteria Assessments: a Comparative Study on MCDA Methods

Aleksandra Bączkiewicz*, Jarosław Wątróbski
Institute of Management, University of Szczecin,
ul. Cukrowa 8, 71-004 Szczecin, Poland
*Doctoral School of University of Szczecin,
ul. Mickiewicza 16, 70-383 Szczecin, Poland
Email: aleksandra.baczkiewicz@phd.usz.edu.pl
jaroslaw.watrobski@usz.edu.pl

Bartłomiej Kizielewicz, Wojciech Sałabun
Research Team on Intelligent Decision Support Systems,
Department of Artificial Intelligence Methods and Applied Mathematics,
Faculty of Computer Science and Information Technology
West Pomeranian University of Technology in Szczecin
ul. Żołnierska 49, 71-210 Szczecin, Poland
Email: {bartlomiej-kizielewicz, wojciech.salabun}@zut.edu.pl

***Abstract*—Objective evaluation in problems considering many, often conflicting criteria is challenging for the decision-maker. This paper presents an approach based on MCDA methods to objectify evaluations in the camera selection problem. The proposed approach includes three MCDA methods, TOPSIS, VIKOR, COMET, and two criterion weighting techniques. Two ranking similarity coefficients were used to compare the resulting rankings of the alternatives: WS and r_w. The performed research confirmed the importance of the appropriate selection of multi-criteria decision-making methods for the solved problem and the relevance of comparative analysis in method selection and construction of objective rankings of alternatives.**

I. Introduction

DEALING with complex, real-world decision-making problems involves recognizing conflicting goals, making decisions with multiple criteria, and aiming for compromise solutions [1], [2]. In response to these requirements, many solutions dedicated to selected areas and general-purpose methods have been developed. Most research has focused on developing and improving new MCDA methods. They differ in many aspects, such as different techniques for determining the weights of criteria in the calculations, the complexity of the algorithms, the way preferences and evaluation criteria are represented, the type of data aggregation and the possibility of considering uncertain data [3].

Despite the existence of many MCDA methods, it is important to be aware that no method is perfect and can be considered suitable for applying to every decision situation or solving every decision problem [4], [5]. In such a condition, it becomes a significant research problem to select a decision support method suitable for the problem under consideration since only a properly selected method can provide a proper solution that reflects the decision maker's preferences [6]. The assessment of alternatives performed using MCDA methods requires considering the decision maker's preferences, which means that the final recommendation may change depending on those preferences [7].

Although there is observed a dynamic development of new MCDA methods and improved existing algorithms, relatively little attention is paid to their proper selection for a given decision problem. Applying the inappropriate method to a particular decision situation can reduce the quality of the recommendation, as different MCDA methods produce inconsistent results. Furthermore, the complexity, unrepeatability, or the fact that decision situations may occur simultaneously over a short time makes their analysis challenging. Consequently, it becomes necessary to apply formal procedures and guidelines for selecting MCDA methods in case of a partial lack of knowledge of the decision situation [8], [9].

Common real-life decision problems in which MCDA methods are applied to solve are issues like the mobile devices selection problem. Among them, there can be considered the mobile phone selection problem, the mobile handset selection problem, laptop selection problem, camera selection problem, where criteria can be features and functionalities such as the size of the in-build camera, battery talk time, brand, colour, camera size and resolution [10].

There are many methods of multi-criteria decision making belonging to different streams. Among them, the two main streams, i.e. the American school and the European school, stand out the most. In addition, there is also an approach that combines elements of both groups and the approach based on a set of rules. Examples of multi-criteria decision-making methods and their assignment to different streams (American, European, mixed or rule-based) are presented using a Table I.

This paper aims to present the study case of an objective camera selection multi-criteria problem. The authors' main objective was to perform a comparative analysis of the results obtained using three selected MCDA methods. Due to the goal of obtaining objective results in an automated process, the authors decided to choose two objective criteria weighting methods. It was assumed that due to the differences in the algorithms included in the MCDA methods, which cause

TABLE I
MULTI-CRITERIA DECISION-MAKING METHODS WITH STREAM AFFILIATIONS AND REFERENCES.

Stream	Acronym	Method Name	References
European	ELECTRE	ELimination Et Choix Traduisant la REalité (ELimination Et Choice Translating REality)	[11]
	PROMETHEE	Preference Ranking Organization METHod for Enrichment of Evaluations	[12]
	TACTIC	Treatment of the Alternatives according To the Importance of Criteria	[13]
American	AHP	Analytic hierarchy process	[14]
	TOPSIS	Technique for the Order of Prioritisation by Similarity to Ideal Solution	[15]
	VIKOR	VIseKriterijumska Optimizacija I Kompromisno Resenje	[16]
	SMART	Simple Mutli-Attribute Rating Technique	[17]
Mixed	IDRA	Intercriteria Decision Rule Approach	[18]
	EVAMIX	Evaluation of Mixed Data	[19]
	PACMAN	Passive and Active Compensability Multicriteria ANalysis	[20]
Rule based	DRSA	Dominance-based rough set approach	[21]
	COMET	Characteristic Objects METhod	[22], [23]

various methods to provide different solutions to the same problems, benchmarking with several methods is an important stage in evaluating a multi-criteria problem. Because MCDA methods are intended to be used in many different fields, the need for a customized approach that considers the particular nature of the problem being analyzed occurs [24]. Using the correlation coefficients of the rankings in the next step allows an objective assessment of the convergence of the rankings and identification of methods that give consistent and outlier results in a specific problem.

For the solution of a described problem, a model-based approach including three MCDA methods, TOPSIS, VIKOR and COMET, has been applied, taking into account two techniques for determining the criteria weights: Mean Weighting, which gives equal weights and Entropy Weighting.

The rest of the paper is organized as follows. Section II provides the preliminaries and main fundamentals of the TOPSIS, VIKOR and COMET methods. In Section III, the study case, including evaluating alternatives and their types, is presented. Section IV shows the results of the performed assessment of alternatives. There is also presented the influence of the methods used in the authors' approach to the outcomes. Section V contains the summary of the conducted survey and conclusions.

II. PRELIMINARIES

A. The TOPSIS Method

The TOPSIS method compares the relative distances between the evaluated alternatives and the positive ideal solution (PIS) and the anti-ideal solution (negative ideal solution - NIS). The goal is to rank the alternatives such that the best alternative is as close as possible to the PIS and as far as possible from the NIS [25]. The TOPSIS method includes the five stages given below [26].

Step 1. Decision matrix is normalized.

In this approach, the greatest and the least values in the considered set are used. The formulas are described as follows (1) and (2):

$$r_{ij} = \frac{x_{ij} - min_j(x_{ij})}{max_j(x_{ij}) - X_{min}} \tag{1}$$

$$r_{ij} = \frac{max_j(x_{ij}) - x_{ij}}{max_j(x_{ij}) - min_j(x_{ij})} \tag{2}$$

Step 2. Weighted values of the normalized decision matrix v_{ij} are determined according to the Equation (3).

$$v_{ij} = w_i r_{ij} \tag{3}$$

Step 3. Calculate the positive ideal solution (PIS) values and negative anti-ideal solution (NIS) vectors. The PIS represented by the vector (4) expresses the maximum values for each criterion, and the NIS is represented by the vector (5) minimum values. It is unnecessary to divide the criteria into cost and profit criteria in this step because the cost criteria were transformed to profit criteria in the normalization step.

$$v_j^+ = \{v_1^+, v_2^+, \ldots, v_n^+\} = \left\{\max_j(v_{ij})\right\} \tag{4}$$

$$v_j^- = \{v_1^-, v_2^-, \ldots, v_n^-\} = \left\{\min_j(v_{ij})\right\} \tag{5}$$

Step 4. Calculate distance from PIS according to the Equation (6) and NIS, using the Equation (7) for each of the alternatives considered [6].

$$D_i^+ = \sqrt{\sum_{j=1}^{n}(v_{ij} - v_j^+)^2} \tag{6}$$

$$D_i^- = \sqrt{\sum_{j=1}^{n}(v_{ij} - v_j^-)^2} \tag{7}$$

Step 5. Calculate the outcome for each of the respected alternatives according to Equation (8). This score takes values between 0 and 1. Thus, the closer the value of a given alternative is to 1, the better is the alternative.

$$C_i = \frac{D_i^-}{D_i^- + D_i^+} \tag{8}$$

B. The VIKOR Method

The VIKOR method (VlseKriterijumska Optimizacija I Kompromisno Resenje), similarly to TOPSIS, takes distance measurement into account, but in this approach, the goal is to identify the alternative closest to the ideal solution. Therefore, the solution sought is a compromise solution [6]. The five steps of the VIKOR method are described below [27], [28], [29].

Step 1. Determinate the best f_j^* and the worst f_j^- value for the function of a particular criterion. For profit criteria, the Equation is used (9).

$$f_j^* = \max_i f_{ij}, \quad f_j^- = \min_i f_{ij} \tag{9}$$

whereas in the case of the cost criteria, the following Equation is used (10).

$$f_j^* = \min_i f_{ij}, \quad f_j^- = \max_i f_{ij} \tag{10}$$

Step 2. Calculate S_i and R_i with using Equations (11) and (12).

$$S_i = \sum_{j=1}^{n} w_j(f_j^* - f_{ij})/(f_j^* - f_j^-) \tag{11}$$

$$R_i = \max_j \left[w_j(f_j^* - f_{ij})/(f_j^* - f_j^-)\right] \tag{12}$$

Step 3. Calculate Q_i with using Equation (13).

$$Q_i = v(S_i - S^*)/(S^- - S^*) + (1-v)(R_i - R^*)/(R^- - R^*) \tag{13}$$

where
$S^* = min_i S_i, \quad S^- = max_i S_i$
$R^* = min_i R_i, \quad R^- = max_i R_i$
v means the weight adopted for the strategy of ”most criteria”.

Step 4. Ranked alternatives S, R and Q are ordered in ascending order. Three ranked lists are the outcome.

Step 5. A compromise solution is proposed considering the conditions of good advantage and acceptable stability within the three vectors obtained in the previous step [29]. The best alternative is the one with the lowest value and the leading position in the ranking Q [30].

C. The COMET Method

The main advantage of the Characteristic Objects METhod (COMET) is its resistance to the rank reversal paradox [31]. COMET method considers fuzzy sets theory. The important steps of this method are the determination and comparison of characteristic objects and the creation of a rule base. Then, each alternative is evaluated in a defuzzification process [25]. The five stages that the COMET method involves are provided below [32], [33], [34].

Step 1. Definition of the space of the problem. The expert determines the dimensionality of the problem with the selection r criteria, $C_1, C_2, ..., C_r$. Then a set of fuzzy numbers is selected for each criterion $C_i, e.g., \left\{\tilde{C}_{i1}, \tilde{C}_{i2}, ..., \tilde{C}_{ic_i}\right\}$ according to the Equation (14).

$$\begin{array}{c} C_1 = \left\{\tilde{C}_{11}, \tilde{C}_{12}, ..., \tilde{C}_{1c_1}\right\} \\ C_2 = \left\{\tilde{C}_{21}, \tilde{C}_{22}, ..., \tilde{C}_{2c_2}\right\} \\ \cdots \\ C_r = \left\{\tilde{C}_{r1}, \tilde{C}_{r2}, ..., \tilde{C}_{rc_r}\right\} \end{array} \tag{14}$$

where $C_1, C_2, ..., C_r$ are the ordinals of the fuzzy numbers for all criteria.

Step 2. The generation of characteristic objects (COs) with the usage of the Cartesian product of the fuzzy numbers' cores of all the criteria according to the Equation (15).

$$CO = \langle C(C_1) \times C(C_2) \times ...C(C_r)\rangle \tag{15}$$

As a result, an ordered set of all COs is obtained (16).

$$\begin{array}{c} CO_1 = \langle C(\tilde{C}_{11}), C(\tilde{C}_{21}), ..., C(\tilde{C}_{r1})\rangle \\ CO_2 = \langle C(\tilde{C}_{11}), C(\tilde{C}_{21}), ..., C(\tilde{C}_{r2})\rangle \\ \cdots \\ CO_t = \langle C(\tilde{C}_{1c_1}), C(\tilde{C}_{2c_2}), ..., C(\tilde{C}_{rc_r})\rangle \end{array} \tag{16}$$

where t is the count of COs and is equal to Equation (17).

$$t = \prod_{i=1}^{r} c_i \tag{17}$$

Step 3. Assessment of characteristic objects by identifying the Matrix of Expert Judgment MEJ by comparing pairwise objects COs by the expert. The MEJ matrix is presented as Equation (18).

$$MEJ = \begin{pmatrix} \alpha_{11} & \alpha_{12} & ... & \alpha_{1t} \\ \alpha_{21} & \alpha_{22} & ... & \alpha_{2t} \\ ... & ... & ... & ... \\ \alpha_{t1} & \alpha_{t2} & ... & \alpha_{tt} \end{pmatrix} \tag{18}$$

where α_{ij} is the outcome of comparing CO_i and CO_j by the expert. The more preferred characteristic object receives a value of 1, and the less preferred object receives a value of 0. If the preferences are equal, both objects get a value of half. This step depends totally on the expert's knowledge and can be represented as (19).

$$\alpha_{ij} = \begin{cases} 0.0, & f_{exp}(CO_i) < f_{exp}(CO_j) \\ 0.5, & f_{exp}(CO_i) = f_{exp}(CO_j) \\ 1.0, & f_{exp}(CO_i) > f_{exp}(CO_j) \end{cases} \tag{19}$$

where the expert function f_{exp} denotes the empirical preferences of the expert.

After the MEJ matrix is provided, a vertical vector of the Summed Judgments SJ is obtained as shown by Equation 20.

$$SJ_i = \sum_{j=1}^{t} \alpha_{ij} \tag{20}$$

Finally, preference values are determined for each characteristic object. As a result, a vertical vector P is obtained, where the i-th row contains the approximate value of preference for CO_i.

Step 4. Each CO and its preference value is converted to a fuzzy rule by using the following Equation (21)

$$IF \quad C\,(\tilde{C}_{1i}) \quad AND \quad C\,(\tilde{C}_{2i}) \quad AND \quad ... \quad THEN \quad P_i \tag{21}$$

In this procedure, a complete fuzzy rule base is prepared.

Step 5. Inference and getting the final ranking. Each alternative is represented as a set of values, e.g. $A_i = \{\alpha_{i1}, \alpha_{i2}, \alpha_{ri}\}$. This set refers to the criteria $C_1, C_2, ..., C_r$. Mamdani's fuzzy inference technique is used to determine the preference of the i-th decision variant. The constant rule base guarantees that the results obtained are unequivocal, which makes COMET completely resistant to the rank reversal paradox [35].

D. Entropy Weighting Method

In the entropy weighting method, the criteria weight is calculated using a measure of uncertainty in the information [36].

Step 1. Normalization of input data using sum normalization method (22).

$$p_{ij} = \frac{x_{ij}}{\sum_{i=1}^{m} x_{ij}} \; i = 1, \ldots, m; \; j = 1, \ldots, n \tag{22}$$

Step 2. Calculation of the entropy value of *j*th criterion using Equation (23).

$$E_j = -\frac{\sum_{i=1}^{m} p_{ij} ln(p_{ij})}{ln(m)} \; j = 1, \ldots, n \tag{23}$$

Step 3. Calculation of the objective weight of *j*th criterion according to the Equation (24).

$$w_j = \frac{1 - E_j}{\sum_{i=1}^{n}(1 - E_j)} \; j = 1, \ldots, n \tag{24}$$

E. Mean Weighting Method

Criteria weights are calculated according to the Equation (25), where n is the number of criteria [37].

$$w_j = 1/n \tag{25}$$

F. Weighted Spearman's Rank Correlation Coefficient

The symmetrical r_w correlation coefficient is calculated by the Equation (26). The sample size is N and x_i and y_i are the positions in rankings which are compared [38].

$$r_w = 1 - \frac{6\sum_{i=1}^{N}(x_i - y_i)^2((N - x_i + 1) + (N - y_i + 1))}{N^4 + N^3 - N^2 - N} \tag{26}$$

G. The WS similarity coefficient

The asymmetrical WS similarity coefficient is calculated according to Equation (27), where N is size of sample and x_i and y_i are the positions in the compared rankings x and y. For this coefficient, changes in the positions at the top of the ranking influence most significantly its value [39].

$$WS = 1 - \sum_{i=1}^{N} 2^{-x_i} \frac{|x_i - y_i|}{max(|x_i - 1|, |x_i - N|)} \tag{27}$$

III. Study Case

This work aimed to study the effect of three different MCDA methods TOPSIS, VIKOR and COMET, on the evaluation results of 20 different camera models. Data on the evaluation criteria values of the selected camera models were obtained from various websites. The selected quantitative criteria represent camera parameters considered by customers during purchase decisions. In modelling decision problems, a very significant issue is determining the importance of decision criteria. There are methods in the literature to obtain the values of criteria weights [38]. In this study, two objective criteria weighting methods were applied: Mean Weighting, which gives equal weights and Entropy Weighting. The selected criteria according to which the alternatives were evaluated are included in Table II. In the next steps of the study, a comparative analysis between the MCDA methods used was performed for each of the criteria weighting methods used. Finally, two ranking correlation coefficients were used to determine the convergence of the obtained rankings: symmetrical r_w and asymmetrical WS.

TABLE II
Selected criteria used in evaluation of camera models

C_i	Name	Type	Unit
C_1	Thickness	Cost	Millimeters $[mm]$
C_2	Width	Cost	Millimeters $[mm]$
C_3	Height	Cost	Millimeters $[mm]$
C_4	Weight	Cost	Gram $[g]$
C_5	Resolution	Profit	Megapixel $[Mpx]$
C_6	4K	Profit	Frames per second $[FPS]$
C_7	FullHD	Profit	Frames per second $[FPS]$
C_8	HD	Profit	Frames per second $[FPS]$
C_9	Viewing angle	Profit	Radian []
C_{10}	Battery life	Profit	Minutes $[min]$
C_{11}	Price	Cost	Polish zloty $[PLN]$

IV. Results and Discussion

The values of each criterion for the alternatives evaluated are given in the decision matrix, displayed in Table III. The decision matrix, normalized by using the Minimum-Maximum normalization method for each weighting technique and MCDA method, is presented in Table IV. For the TOPSIS and COMET methods, the best alternative is the alternative that scored the highest preference value. Therefore, the lower the preference value, the lower the alternative is ranked. For the VIKOR method, the opposite is true. In its case,

TABLE III
THE PERFORMANCE TABLE OF THE ALTERNATIVES $A_1 - A_{20}$.

Alternatives	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}
SONY FDR-X3000	29.40	83.00	47.00	114	12.0	30	120	240	170	90	1717.75
DJI Pocket 2 Creator Combo	30.00	38.10	124.70	117	16.0	60	60	60	93	70	2389.00
GÖTZE & JENSEN S-Line SC501	29.28	59.27	41.13	58	16.0	30	60	120	170	78	239.99
GOPRO HERO9	33.60	71.00	55.00	159	23.6	60	240	240	132	140	2099.00
Xblitz Move 4K+	21.00	59.00	41.00	66	16.0	24	60	120	170	70	439.00
DJI Osmo Action	35.00	65.00	42.00	134	12.0	60	240	240	145	60	1087.00
Insta360 ONE R-1-Inch Edition	47.00	79.00	54.00	158	19.0	60	120	120	360	72	2499.00
GOPRO HERO7	28.30	62.30	44.90	116	12.0	30	60	60	130	90	999.99
DJI Osmo Pocket	36.90	28.60	121.60	130	12.0	60	120	120	80	80	1099.00
GOXTREME Enduro	32.00	59.00	41.00	60	16.0	30	120	120	170	60	302.96
GOPRO HERO8	28.40	66.30	48.60	126	12.0	60	240	240	132	135	1629.00
Insta360 One X2	29.80	46.00	113.00	47	18.0	50	50	50	360	72	2099.00
SJCAM A20	20.20	64.00	80.00	70	8.0	24	60	120	166	480	699.99
LAMAX X9.1	33.00	60.00	44.00	59	12.0	30	60	120	170	90	388.00
MANTA MM9259	29.00	59.00	41.00	55	16.0	30	60	120	170	120	299.00
SJCAM SJ4000 WiFi	29.00	59.00	41.00	182	12.0	30	30	60	94	140	249.00
LAMAX Action X3.1 Atlas	29.80	59.20	41.00	65	16.0	30	60	120	160	90	219.99
SJCAM SJ10 Pro	28.80	62.50	41.00	70	12.0	60	120	120	170	138	1399.99
GOXTREME Pioneer	24.00	40.00	59.00	60	12.0	10	30	30	140	78	269.99
TRACER eXplore SJ 4561	30.00	60.00	45.00	201	16.0	30	30	30	170	90	199.99

TABLE IV
NORMALIZED DECISION MATRIX

A_i	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}
A_1	0.3745	0.0000	0.6231	0.4328	0.5085	0.5000	0.5000	1.0000	0.4722	0.1875	0.3126
A_2	0.3617	0.5410	0.0000	0.4179	0.6780	1.0000	0.2500	0.2500	0.2583	0.1458	0.0440
A_3	0.3770	0.2859	0.6702	0.7114	0.6780	0.5000	0.2500	0.5000	0.4722	0.1625	0.9040
A_4	0.2851	0.1446	0.5589	0.2090	1.0000	1.0000	1.0000	1.0000	0.3667	0.2917	0.1601
A_5	0.5532	0.2892	0.6712	0.6716	0.6780	0.4000	0.2500	0.5000	0.4722	0.1458	0.8243
A_6	0.2553	0.2169	0.6632	0.3333	0.5085	1.0000	1.0000	1.0000	0.4028	0.1250	0.5650
A_7	0.0000	0.0482	0.5670	0.2139	0.8051	1.0000	0.5000	0.5000	1.0000	0.1500	0.0000
A_8	0.3979	0.2494	0.6399	0.4229	0.5085	0.5000	0.2500	0.2500	0.3611	0.1875	0.5998
A_9	0.2149	0.6554	0.0249	0.3532	0.5085	1.0000	0.5000	0.5000	0.2222	0.1667	0.5602
A_{10}	0.3191	0.2892	0.6712	0.7015	0.6780	0.5000	0.5000	0.5000	0.4722	0.1250	0.8788
A_{11}	0.3957	0.2012	0.6103	0.3731	0.5085	1.0000	1.0000	1.0000	0.3667	0.2812	0.3481
A_{12}	0.3660	0.4458	0.0938	0.7662	0.7627	0.8333	0.2083	0.2083	1.0000	0.1500	0.1601
A_{13}	0.5702	0.2289	0.3585	0.6517	0.3390	0.4000	0.2500	0.5000	0.4611	1.0000	0.7199
A_{14}	0.2979	0.2771	0.6472	0.7065	0.5085	0.5000	0.2500	0.5000	0.4722	0.1875	0.8447
A_{15}	0.3830	0.2892	0.6712	0.7264	0.6780	0.5000	0.2500	0.5000	0.4722	0.2500	0.8804
A_{16}	0.3830	0.2892	0.6712	0.0945	0.5085	0.5000	0.1250	0.2500	0.2611	0.2917	0.9004
A_{17}	0.3660	0.2867	0.6712	0.6766	0.6780	0.5000	0.2500	0.5000	0.4444	0.1875	0.9120
A_{18}	0.3872	0.2470	0.6712	0.6517	0.5085	1.0000	0.5000	0.5000	0.4722	0.2875	0.4398
A_{19}	0.4894	0.5181	0.5269	0.7015	0.5085	0.1667	0.1250	0.1250	0.3889	0.1625	0.8920
A_{20}	0.3617	0.2771	0.6391	0.0000	0.6780	0.5000	0.1250	0.1250	0.4722	0.1875	0.9200

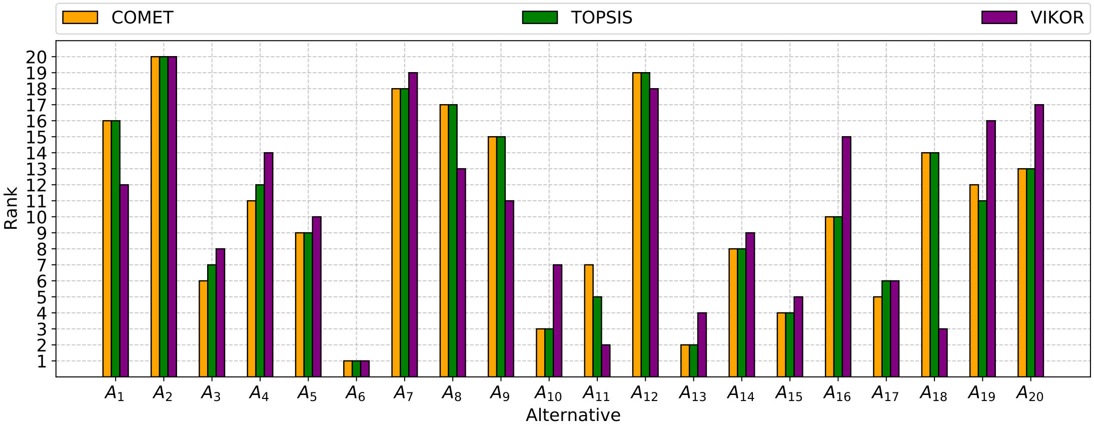


Fig. 1. Comparison of rankings received with using Entropy Weighting for TOPSIS, VIKOR and COMET

TABLE V
PREFERENCE VALUES AND RANKINGS OBTAINED WITH USING ENTROPY WEIGHTING FOR TOPSIS, VIKOR AND COMET

A_i	$COMET_{pref}$	$TOPSIS_{pref}$	$VIKOR_{pref}$	$COMET_{rank}$	$TOPSIS_{rank}$	$VIKOR_{rank}$
A_1	0.3902	0.4287	0.3557	16	16	12
A_2	0.0848	0.2060	0.9500	20	20	20
A_3	0.5901	0.5424	0.2118	6	7	8
A_4	0.4709	0.4853	0.4203	11	12	14
A_5	0.5385	0.5176	0.2609	9	9	10
A_6	0.6677	0.5885	0.1240	1	1	1
A_7	0.1787	0.3136	0.8273	18	18	19
A_8	0.3827	0.4192	0.3778	17	17	13
A_9	0.4278	0.4459	0.3531	15	15	11
A_{10}	0.6377	0.5703	0.2029	3	3	7
A_{11}	0.5796	0.5463	0.1397	7	5	2
A_{12}	0.1631	0.2806	0.6954	19	19	18
A_{13}	0.6489	0.5877	0.1506	2	2	4
A_{14}	0.5619	0.5302	0.2232	8	8	9
A_{15}	0.5988	0.5494	0.1856	4	4	5
A_{16}	0.4969	0.4978	0.4415	10	10	15
A_{17}	0.5964	0.5459	0.1976	5	6	6
A_{18}	0.4580	0.4682	0.1433	14	14	3
A_{19}	0.4687	0.4856	0.4451	12	11	16
A_{20}	0.4641	0.4850	0.4674	13	13	17

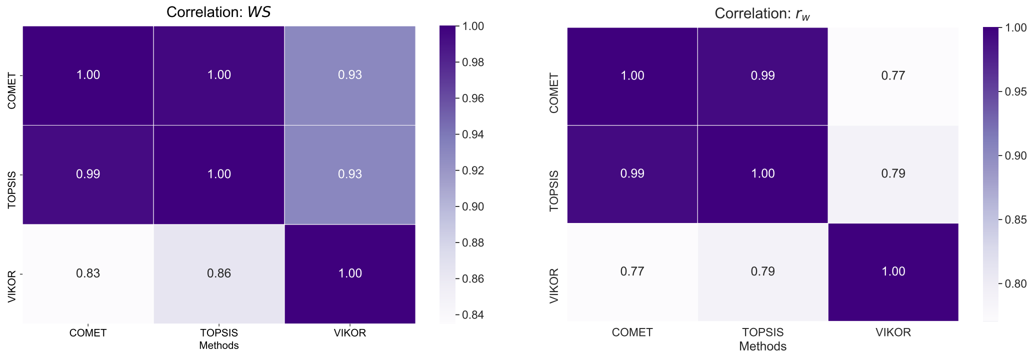


Fig. 2. WS and r_w correlation heat maps for TOPSIS, VIKOR and COMET with using Entropy Weighting

TABLE VI
PREFERENCE VALUES AND RANKINGS OBTAINED WITH USING MEAN WEIGHTING (EQUAL WEIGHTS) FOR TOPSIS, VIKOR AND COMET

A_i	$COMET_{pref}$	$TOPSIS_{pref}$	$VIKOR_{pref}$	$COMET_{rank}$	$TOPSIS_{rank}$	$VIKOR_{rank}$
A_1	0.4516	0.4646	0.7887	13	13	14
A_2	0.3565	0.4042	1.0000	20	20	20
A_3	0.5323	0.5233	0.4939	8	8	6
A_4	0.5797	0.5516	0.0356	2	2	2
A_5	0.5358	0.5249	0.5375	7	7	9
A_6	0.5731	0.5487	0.5186	3	3	8
A_7	0.4353	0.4590	0.8249	15	15	16
A_8	0.4025	0.4248	0.7055	17	18	13
A_9	0.4223	0.4462	0.8539	16	16	17
A_{10}	0.5421	0.5315	0.5875	6	5	10
A_{11}	0.5815	0.5565	0.0192	1	1	1
A_{12}	0.4814	0.4878	0.6455	12	12	11
A_{13}	0.5159	0.5111	0.6457	10	10	12
A_{14}	0.4882	0.4913	0.5150	11	11	7
A_{15}	0.5424	0.5313	0.2023	5	6	4
A_{16}	0.3868	0.4232	0.9327	19	19	19
A_{17}	0.5268	0.5195	0.4292	9	9	5
A_{18}	0.5459	0.5351	0.0790	4	4	3
A_{19}	0.4393	0.4630	0.8160	14	14	15
A_{20}	0.3923	0.4281	0.9205	18	17	18

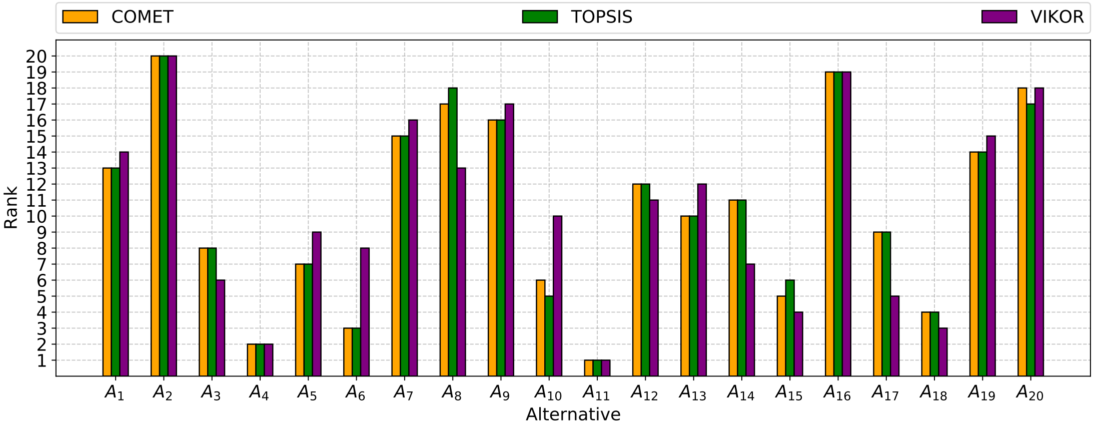


Fig. 3. Comparison of rankings received with using Mean Weighting (Equal Weights) for TOPSIS, VIKOR and COMET

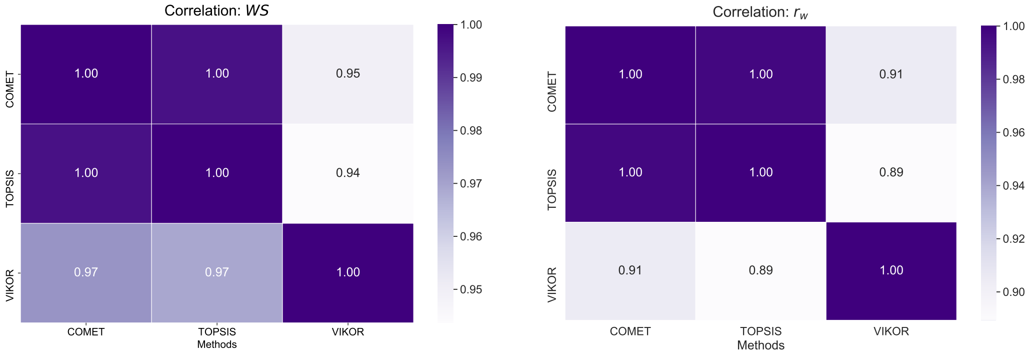


Fig. 4. WS and r_w correlation heat maps for TOPSIS, VIKOR and COMET with using Mean Weighting (Equal Weights)

the best alternative is the alternative for which the lowest preference value was calculated. Therefore, as the preference value increases, the alternative decreases in ranking.

A. *Results for Entropy Weighting*

Preference values and rankings obtained for each MCDA method with applying entropy weights are contained in Table V. Comparison of rankings is visualized in Figure 1. In the rankings obtained using entropy weights, only two alternatives ranked equally in the rankings created by the MCDA methods applied. These include A_6, which is the leader in all three rankings, and A_2, which is always last. Analysis of the obtained rankings allows us to conclude that the VIKOR method has the most significant impact on the differences in rankings. Ranking received using this method demonstrates the most divergent values (range of differences including four positions for A_1, A_8, A_9, A_{10}, A_{20}, five positions for A_{11}, A_{16}, and eleven positions for A_{18}). The rankings obtained using TOPSIS and COMET methods show very high convergence. As many as 15 alternatives are in identical positions. For the remaining alternatives, the differences are minimal (one position for A_3, A_4, A_{17}, A_{19} and two positions for A_{11}).

The similarity between the rankings obtained using each MCDA method was then examined. Results of investigation of rankings' similarity are displayed in Figure 2. Two ranking similarity coefficients were used to investigate the correlation: WS and r_w. The highest value of asymmetrical WS coefficient was noticed for COMET and TOPSIS (1.00) and TOPSIS and COMET (0.99). The lower value was received for COMET and VIKOR and TOPSIS and VIKOR (0.93). The lowest correlation was observed for VIKOR and TOPSIS (0.96) and VIKOR and COMET (0.83).

When investigating the similarity of rankings using the r_w coefficient, the highest correlation was found for COMET and TOPSIS (0.99), lower for TOPSIS and VIKOR (0.79), and lowest for VIKOR and COMET (0.77). Thus, the ranking similarity examination results confirm the outliers in the ranking achieved by the VIKOR method.

B. *Results for Mean Weighting*

Preference values for TOPSIS, VIKOR and COMET with applying equal weights are contained in Table VI. Comparison of rankings is illustrated in Figure 3. The same rankings for the three MCDA methods were obtained for only four alternatives when Mean Weighting was used. Among them are A_{11}, which is the ranking leader, A_4 in second place, A_{16} in second-to-last place, and A_2 in the last place. Thus, another alternative is the leader for Mean Weighting than Entropy Weighting. The most significant differences between the obtained rankings were observed for the VIKOR method (range of differences including five positions for A_6, A_8, A_{10}, and four positions for A_{14} and A_{17}). On the other hand, for Mean Weighting, the rankings obtained with the TOPSIS and COMET methods were the most consistent. The rankings were identical for as many as 16 alternatives, while for four alternatives (A_8, A_{10}, A_{15}, A_{20}), the differences included only one position.

Values of ranking similarity coefficients are displayed in Figure 4. In the ranking similarity study, the highest WS value was obtained for COMET and TOPSIS (1.00), followed by VIKOR and COMET and VIKOR and TOPSIS (0.97), and the lowest for COMET and VIKOR (0.95) and TOPSIS and VIKOR (0.94). The highest r_w value was received for COMET and TOPSIS (1.00), lower for VIKOR and COMET (0.91), and lowest for TOPSIS and VIKOR (0.89).

The results of the performed research demonstrate that the complexity of decision problems containing many different criteria makes it difficult to identify a universal method to obtain the best solution for various problems. Therefore, when there is a need to obtain an objective solution to multi-criteria decision problems, hybrid approaches, in which different algorithms are combined to solve the decision problem, seem to be suitable [40]. A well-known example is the hybrid DSS 3.0 system proposed by Budzinski and Becker. In this system, the values of criteria weights are determined by the AHP method, while the ELECTRE Tri method is used to create the ranking [41]. The described hybrid approach is worth attention and consideration in further research directions.

V. Conclusions

This paper aimed to investigate the effect of selected MCDA methods and objective weighting techniques on the objectivity of the resulting rankings. The case study in this work was the camera selection problem. The results obtained confirm that several conditions must be respected to obtain appropriate assessment results using MCDA methods. First, it is essential to select methods for the problem to be adequately solved. Second, benchmarking with other methods allowing for comparative analysis is required. Also, a proper selection of criteria weights that reflect the preferences of the decision-maker is recommended.

The study shows that the most comparable rankings were achieved using TOPSIS and COMET methods. Outlier results of the VIKOR method contribute to the disturbance of objectivity of received results. Due to the careful selection of several MCDA methods and the comparative analysis performed, it was possible to determine a set of methods providing convergent and objective results. Obtained results encourage continuing the research with other MCDA methods to extend the set of methods enabling objectivization of evaluations in the undertaken problem.

Acknowledgment

The work was supported by the project financed within the framework of the program of the Minister of Science and Higher Education under the name “Regional Excellence Initiative” in the years 2019–2022, Project Number 001/RID/2018/19; the amount of financing: PLN 10.684.000,00 (A.B. and J.W.) and by the National Science Centre, Decision number UMO-2018/29/B/HS4/02725 (B.K. and W.S.).

References

[1] B. Kizielewicz, J. Wątróbski, and W. Sałabun, “Identification of Relevant Criteria Set in the MCDA Process—Wind Farm Location Case Study,” *Energies*, vol. 13, no. 24, p. 6548, 2020. doi: https://doi.org/10.3390/en13246548

[2] J. Wątróbski, P. Ziemba, and W. Wolski, “MCDA-based decision support system for sustainable management-RES case study,” in *2016 Federated Conference on Computer Science and Information Systems (FedCSIS), Gdansk, Poland, 11–14 September 2016*. IEEE, 2016. doi: https://doi.org/10.15439/2016F489 pp. 1235–1239.

[3] C. Beaudrie, C. J. Corbett, T. A. Lewandowski, T. Malloy, and X. Zhou, “Evaluating the Application of Decision Analysis Methods in Simulated Alternatives Assessment Case Studies: Potential Benefits and Challenges of using MCDA,” *Integrated Environmental Assessment and Management*, vol. 17, no. 1, pp. 27–41, 2021. doi: https://doi.org/10.1002/ieam.4316

[4] A. Karczmarczyk, J. Wątróbski, and J. Jankowski, “Comparative Study of Different MCDA-Based Approaches in Sustainable Supplier Selection Problem,” in *Information Technology for Management: Emerging Research and Applications*. Springer, 2018. doi: https://doi.org/10.1007/978-3-030-15154-6_10 pp. 176–193.

[5] N. Tsotsolas and S. Alexopoulos, “MCDA Approaches for Efficient Strategic Decision Making,” in *Preference Disaggregation in Multiple Criteria Decision Analysis*. Springer, 2018. doi: https://doi.org/10.1007/978-3-319-90599-0_2 pp. 17–58.

[6] W. Sałabun, J. Wątróbski, and A. Shekhovtsov, “Are MCDA Methods Benchmarkable? A Comparative Study of TOPSIS, VIKOR, COPRAS, and PROMETHEE II Methods,” *Symmetry*, vol. 12, no. 9, p. 1549, 2020. doi: https://doi.org/10.3390/sym12091549

[7] M. Cinelli, M. Kadziński, M. Gonzalez, and R. Słowiński, “How to support the application of multiple criteria decision analysis? Let us start with a comprehensive taxonomy,” *Omega*, p. 102261, 2020. doi: https://doi.org/10.1016/j.omega.2020.102261

[8] A. Papapostolou, F. D. Mexis, E. Sarmas, C. Karakosta, and J. Psarras, “Web-based Application for Screening Energy Efficiency Investments: A MCDA Approach,” in *2020 11th International Conference on Information, Intelligence, Systems and Applications (IISA), Piraeus, Greece, 15–17 July 2020*. IEEE, 2020. doi: https://doi.org/10.1109/IISA50023.2020.9284403 pp. 1–7.

[9] J. Wątróbski, J. Jankowski, P. Ziemba, A. Karczmarczyk, and M. Zioło, “Generalised framework for multi-criteria method selection,” *Omega*, vol. 86, pp. 107–124, 2019. doi: https://doi.org/10.1016/j.omega.2018.07.004

[10] R. K. Dhurkari, “MCGL: a new reference dependent MCDM method,” *International Journal of Operational Research*, vol. 36, no. 4, pp. 477–495, 2019. doi: https://doi.org/10.1504/IJOR.2019.104053

[11] B. Roy, “Classement et choix en présence de points de vue multiples,” *Revue française d’informatique et de recherche opérationnelle*, vol. 2, no. 8, pp. 57–75, 1968. doi: http://www.numdam.org/item?id=RO_1968__2_1_57_0

[12] J.-P. Brans and P. Vincke, “Note—a preference ranking organisation method: (the promethee method for multiple criteria decision-making),” *Management science*, vol. 31, no. 6, pp. 647–656, 1985. doi: https://doi.org/10.1287/mnsc.31.6.647

[13] J.-M. Martel and B. Matarazzo, “Other outranking approaches,” in *Multiple criteria decision analysis: state of the art surveys*. Springer, 2005. doi: https://doi.org/10.1007/978-1-4939-3094-4_7 pp. 197–259.

[14] A. Darko, A. P. C. Chan, E. E. Ameyaw, E. K. Owusu, E. Pärn, and D. J. Edwards, "Review of application of analytic hierarchy process (ahp) in construction," *International journal of construction management*, vol. 19, no. 5, pp. 436–452, 2019. doi: https://doi.org/10.1080/15623599.2018.1452098

[15] C.-L. Hwang and K. Yoon, "Methods for multiple attribute decision making," in *Multiple attribute decision making*. Springer, 1981. doi: https://doi.org/10.1007/978-3-642-48318-9_3 pp. 58–191.

[16] L. Duckstein and S. Opricovic, "Multiobjective optimization in river basin development," *Water resources research*, vol. 16, no. 1, pp. 14–20, 1980. doi: https://doi.org/10.1029/WR016i001p00014

[17] N. Sinha, N. Priyanka, and P. Joshi, "Using spatial multi-criteria analysis and ranking tool (SMART) in earthquake risk assessment: A case study of Delhi region, India," *Geomatics, Natural Hazards and Risk*, vol. 7, no. 2, pp. 680–701, 2016. doi: https://doi.org/10.1080/19475705.2014.945100

[18] S. Greco, "A new PCCA method: Idra," *European Journal of Operational Research*, vol. 98, no. 3, pp. 587–601, 1997. doi: https://doi.org/10.1016/S0377-2217%2896%2900022-7

[19] H. Voogd, "Multicriteria evaluation with mixed qualitative and quantitative data," *Environment and Planning B: Planning and Design*, vol. 9, no. 2, pp. 221–236, 1982. doi: https://doi.org/10.1068/b090221

[20] A. Giarlotta, "Passive and active compensability multicriteria annalysis (PACMAN)," *Journal of Multi-Criteria Decision Analysis*, vol. 7, no. 4, pp. 204–216, 1998. doi: https://doi.org/10.1002/%28SICI%291099-1360%28199807%297:4%3C204::AID-MCDA192%3E3.0.CO%3b2-5

[21] P. Fortemps, S. Greco, and R. Słowiński, "Multicriteria choice and ranking using decision rules induced from rough approximation of graded preference relations," in *International Conference on Rough Sets and Current Trends in Computing, Uppsala, Sweden, 1–5 June 2004*. Springer, 2004. doi: https://doi.org/10.1007/978-3-540-25929-9_62 pp. 510–522.

[22] W. Sałabun, "The Characteristic Objects Method: A New Distance-based Approach to Multicriteria Decision-making Problems," *Journal of Multi-Criteria Decision Analysis*, vol. 22, no. 1-2, pp. 37–50, 2015. doi: https://doi.org/10.1002/mcda.1525

[23] W. Sałabun and A. Piegat, "Comparative analysis of MCDM methods for the assessment of mortality in patients with acute coronary syndrome," *Artificial Intelligence Review*, vol. 48, no. 4, pp. 557–571, 2017. doi: https://doi.org/10.1007/s10462-016-9511-9

[24] J. Wątróbski and J. Jankowski, "Knowledge management in MCDA domain," in *2015 Federated Conference on Computer Science and Information Systems (FedCSIS), Lodz, Poland, 13–16 September 2015*. IEEE, 2015. doi: https://doi.org/10.15439/2015F295 pp. 1445–1450.

[25] A. Karczmarczyk, J. Wątróbski, G. Ladorucki, and J. Jankowski, "MCDA-based approach to sustainable supplier selection," in *2018 Federated Conference on Computer Science and Information Systems (FedCSIS), Poznan, Poland, 9–12 September 2018*. IEEE, 2018. doi: https://doi.org/10.15439/2018F336 pp. 769–778.

[26] V. Yadav, S. Karmakar, P. P. Kalbar, and A. K. Dikshit, "PyTOPS: A Python based tool for TOPSIS," *SoftwareX*, vol. 9, pp. 217–222, 2019. doi: https://doi.org/10.1016/j.softx.2019.02.004

[27] T. Imandasari, M. G. Sadewo, A. P. Windarto, A. Wanto, H. O. L. Wijaya, and R. Kurniawan, "Analysis of the Selection Factor of Online Transportation in the VIKOR Method in Pematangsiantar city," in *Journal of Physics: Conference Series, Niagara Hotel, Parapat, Indonesia, 10–12 October 2018*, vol. 1255, no. 1. IOP Publishing, 2019. doi: https://doi.org/10.1088/1742-6596/1255/1/012008 p. 012008.

[28] A. Mardani, E. K. Zavadskas, K. Govindan, A. Amat Senin, and A. Jusoh, "VIKOR technique: A systematic review of the state of the art literature on methodologies and applications," *Sustainability*, vol. 8, no. 1, p. 37, 2016. doi: https://doi.org/10.3390/su8010037

[29] D. Siregar, H. Nurdiyanto, S. Sriadhi, D. Suita, U. Khair, R. Rahim, D. Napitupulu, A. Fauzi, A. Hasibuan, M. Mesran *et al.*, "Multi-attribute decision making with VIKOR method for any purpose decision," in *Journal of Physics: Conference Series, Kuching, Sarawak, Malaysia, 25–27 November 2017*, vol. 1019, no. 1. IOP Publishing, 2018. doi: https://doi.org/10.1088/1742-6596/1019/1/012034 p. 012034.

[30] M. Kumar and C. Samuel, "Selection of best renewable energy source by using VIKOR method," *Technology and Economics of Smart Grids and Sustainable Energy*, vol. 2, no. 1, p. 8, 2017. doi: https://doi.org/10.1007/s40866-017-0024-7

[31] J. Wątróbski, W. Sałabun, A. Karczmarczyk, and W. Wolski, "Sustainable decision-making using the COMET method: An empirical study of the ammonium nitrate transport management," in *2017 Federated Conference on Computer Science and Information Systems (FedCSIS), Prague, Czech Republic, 3–6 September 2017*. IEEE, 2017. doi: https://doi.org/10.15439/2017F455 pp. 949–958.

[32] W. Sałabun, J. Wątróbski, and A. Piegat, "Identification of a multi-criteria model of location assessment for renewable energy sources," in *International Conference on Artificial Intelligence and Soft Computing, Zakopane, Poland, 12–16 June 2016*. Springer, 2016. doi: https://doi.org/10.1007/978-3-319-39378-0_28 pp. 321–332.

[33] W. Sałabun, P. Ziemba, and J. Wątróbski, "The rank reversals paradox in management decisions: The comparison of the AHP and COMET methods," in *International Conference on Intelligent Decision Technologies, Tenerife, Spain, 15–17 June 2016*. Springer, 2016. doi: https://doi.org/10.1007/978-3-319-39630-9_15 pp. 181–191.

[34] W. Sałabun, A. Karczmarczyk, J. Wątróbski, and J. Jankowski, "Handling data uncertainty in decision making with COMET," in *2018 IEEE Symposium Series on Computational Intelligence (SSCI), Bangalore, India, 18–21 November 2018*. IEEE, 2018. doi: https://doi.org/10.1109/SSCI.2018.8628934 pp. 1478–1484.

[35] A. Shekhovtsov, J. Kołodziejczyk, and W. Sałabun, "Fuzzy Model Identification Using Monolithic and Structured Approaches in Decision Problems with Partially Incomplete Data," *Symmetry*, vol. 12, no. 9, p. 1541, 2020. doi: https://doi.org/10.3390/sym12091541

[36] H. Li, W. Wang, L. Fan, Q. Li, and X. Chen, "A novel hybrid MCDM model for machine tool selection using fuzzy DEMATEL, entropy weighting and later defuzzification VIKOR," *Applied Soft Computing*, vol. 91, p. 106207, 2020. doi: https://doi.org/10.1016/j.asoc.2020.106207

[37] N. Yalcin and U. Ünlü, "A multi-criteria performance analysis of Initial Public Offering (IPO) firms using CRITIC and VIKOR methods," *Technological and Economic development of Economy*, vol. 24, no. 2, pp. 534–560, 2018. doi: http://dx.doi.org/10.3846/20294913.2016.1213201

[38] A. Shekhovtsov, V. Kozlov, V. Nosov, and W. Sałabun, "Efficiency of Methods for Determining the Relevance of Criteria in Sustainable Transport Problems: A Comparative Case Study," *Sustainability*, vol. 12, no. 19, p. 7915, 2020. doi: https://doi.org/10.3390/su12197915

[39] W. Sałabun and K. Urbaniak, "A new coefficient of rankings similarity in decision-making problems," in *International Conference on Computational Science, Amsterdam, The Netherlands, 3–5 June 2020*. Springer, 2020. doi: https://doi.org/10.1007/978-3-030-50417-5_47 pp. 632–645.

[40] J. Becker and R. Budziński, "Optimization Procedure of the Multi-parameter Assessment and Bidding of Decision-Making Variants in the Computerized Decision Support System," in *Computational Collective Intelligence*. Springer, 2015. doi: https://doi.org/10.1007/978-3-319-24306-1_18 pp. 182–192.

[41] L. Fabisiak, R. Budziński, K. Szczypiór-Piasecka, and P. Ziętek, "Zastosowanie metody wielokryterialnej do analizy diagnostycznej pacjenta z chorobą zwyrodnieniową stawu biodrowego," *Studia Informatica Pomerania*, no. 4 (42), pp. 15–25, 2016. doi: http://dx.doi.org/10.18276/si.2016.42-02

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 427–436

DOI: 10.15439/2021F122
ISSN 2300-5963 ACSIS, Vol. 25

Critical Success Factors for Digitalization Projects

Christian Leyh, Konstanze Köppel
Technische Universität Dresden, Chair of Information Systems, esp. IS in Manufacturing and Commerce
Helmholtzstr. 10, 01069 Dresden, Germany
Email: christian.leyh@tu-dresden.de

Sarah Neuschl, Milan Pentrack
Fraunhofer Center for International Management and Knowledge Economy IMW
Neumarkt 9-19, 04109 Leipzig, Germany
Email: sarah.neuschl@imw.fraunhofer.de, milan.pentrack@imw.fraunhofer.de

Abstract—**Our paper provides insights into which critical success factors (CSFs) for digitalization projects are seen as important from the companies' perspective based on an online survey. The results presented in this paper show that CSFs of the dimensions of *Corporate organization* and *Technology* are considered to be of particular relevance, as stated by the companies, with *Corporate culture*, *Top management support*, and a *Unified digital corporate strategy / vision* as the three most important CSFs. Therefore, this paper contributes to the CSF research regarding *digital transformation* and enables the development of practice-oriented recommendations for action and assistance in shaping digital transformation.**

I. Motivation

TODAY, more than ever, society is undergoing a rapidly evolving digital transformation: government institutions, households, enterprises, and their interactions are all changing as a result of the increasing prevalence and rapidly growing potential of digital technologies. "It is not too much of a stretch to think we have entered a golden age of digital innovation. Owing to the 50-year march of Moore's Law, we have witnessed the creation of a relatively cheap and increasingly easy-to-use world-wide digital infrastructure of computers, mobile devices, broadband network connections, and advanced application platforms" [1]. For companies, in particular, being able to rely on a deep understanding of information technology (IT), in general, and digital innovation, in particular, has never been more important. The technological possibilities, especially concerning the merging of the physical with the digital world, are leading to fundamental paradigm shifts that affect all industries. Nowadays, enterprises have to participate in global digital networking, improve automation of business processes, and reengineer existing business models to gain momentum in digital innovation. Furthermore, the progressive and steady digitalization of society itself, with associated changes, is also playing a role in the daily lives of enterprises. The consequences of this development and the question of whether these changes should be seen as positive or negative are omnipresent [2]–[6]. Digitalization has long since ceased to be a mere buzzword but has rather become a strategic competitive factor. Moreover, digitalization is often seen as an *enabler* to increase resilience in companies. Here, the positive effects of digital technologies and business models are emphasized. The COVID-19 crisis lends new relevance to this thesis, as many companies were only able to maintain certain processes with the help of digital tools (e.g., video conferencing, remote services) [7]–[9].

The imperative came up that companies should use the current pandemic as another starting point or leverage for digital transformation as well as for structural change [10]. In the COVID-19 crisis, it became particularly apparent that the challenge is not merely the *implementation and use of digital technologies*, since the accompanying appropriate *changes at every organizational level*, e.g., business process adjustments, business model innovations, and restructuring the company organization itself, are at least of equal importance. Consequently, mastering the challenges posed by digitalization has long since ceased to be merely the task of the IT department but rather the entire company [6].

Activities and projects in digital transformation are usually highly complex and time-intensive, thus leading to great opportunities for companies as well as enormous risks. To avoid being "swallowed up" by the risks, it is imperative for companies to focus on the factors that influence digitalization projects. In this context, various studies (e.g., [11]–[17]) have shown that paying attention to these so-called "critical success factors" (CSFs) can have a positive influence on the success of IT projects and their subsequent use, thus minimizing the project risks.

In both scientific and practice-oriented literature, the CSFs for digitalization projects are primarily discussed against the background of the difference between digitalization projects and "classic IT projects." Taking up this discussion and topic, we set up a long-term research project at the Chair of Information Systems, esp. IS in Manufacturing and Commerce at Technische Universität Dresden that specifically addresses *CSFs influencing projects in the context of the digital transformation of enterprises*. In a first step, we conducted an extensive systematic literature analysis to identify the CSFs of digitalization projects. Second, we set up an interview study with several selected companies to verify the factors identified in the literature and identify additional factors (see [18]). This resulted in 25 CSFs that form the basis of the third step in the research project and, thus, the basis of this paper. The aim of this third step is to examine the importance/relevance of the identified 25 CSFs for digitalization projects with a quantitative study using an online survey. Furthermore, we aim to examine the implementation and characteristics of these factors in the companies' projects as a fourth step (which will not be part of this paper). For the third step, we derived four research questions to guide our analysis. In the following, we will only focus on the central research question for the aim and scope of this paper:

Which critical success factors are considered (particularly) important in digitalization projects?

Taking up this research question, this paper aims to provide initial answers by presenting and discussing selected results of the online survey. To this end, we structured the paper as follows. This introduction is followed by a brief overview of the theoretical foundations of our study. Afterwards, we present the design of our study and the structure of the questionnaire. Then we describe selected results of the survey. Finally, the paper concludes with a discussion and conclusion with an outlook on further research steps.

II. Theoretical Background

A. Digital Transformation

Digital transformation is the inner engine of a highly extensive transformation, as the effects of which are technologically detectable but the overall consequences for the economy and society are not traceable. Driven by the fourth industrial revolution, it is not only customer behavior that changed but also the way people, organizations, and industries interact with each other [19]. So far, there is no universal definition of digital transformation in the literature. The terms *digital transformation*, *digitalization*, and *digital age* are frequently used as synonyms. Therefore, we use the term *digital transformation (DT)* in this study. Despite the different views of DT, we can see that DT is a development driven by digital technologies and constant changes in society as well as companies. DT is described as linking together the changes in strategies, business models, cultures, structures, and processes in companies with the goal of strengthening the company's market position using digital technologies [20]. Furthermore, DT differs from a classic change process based on three specific characteristics: The first characteristic is that DT often starts with the customer. Here, digital customer data, in particular, play a central role. New business models, for example, can emerge from this resource. The second characteristic is that DT represents more than just the optimization of business processes and IT. In general, DT encompasses the complete renewal of the entire business model. The third and final characteristic shows that DT is an open-ended and long-term process. The most profound difference between DT and a classic change process is its open-endedness. It fosters a completely new kind of management challenge, since there have been little to no standards or best practices that companies can draw on for help. Management must start from new premises for the conception and implementation of DT processes [21].

In conclusion, there is no clear definition for DT in science and practice, as various definitions represent DT in a general or highly simplified way. In the context of our study, we define DT as follows:

DT refers to the fundamental transformation of society as well as the economy using digital technologies. DT not only has social, cultural, legal, and political implications but also consequences for all corporate structures and value chains. For companies to master DT successfully, new business models, strategies, organizational forms, and processes are necessary, as well as a strong customer-centricity.

B. Digitalization Projects

DT is leading a shift in many companies. The transformation of the company in DT is often traversed in several digitalization projects. However, there is no uniform definition for digitalization projects in the literature. In general, a digitalization project is a project that pursues the goal to digitally transform a specific area of the company. This can involve not only redesigning parts of the working environment but also networking systems or production facilities through machines. In most cases, the benefits of the specific digitalization project for the employee or customer are unclear at the project's start (as they can only be estimated at this point) and are, therefore, overshadowed by fears. This is because many target groups have not yet had any experience with such digitalization projects and are, therefore, unable to assess their future impact. Ignorance and uncertainty are often the biggest hurdles in the implementation of digitalization projects. In DT, companies have a particularly difficult time, because such changes are not only linked to large investments but also to adjustments within the organization in the areas of responsibility and leadership behavior [22]. In any digitalization project, it is important to consider the reservations, wishes, and goals of the various target groups. In general, four phases divide the procedure of a digitalization project: *goal setting*, *strengthening project acceptance*, *implementation*, and *control*. The first phase derives the objectives and strategies for the DT project. Since this forms the basis of the entire digitalization project, it is essential to involve all target groups. In the second phase, a strategic and tactical concept design of the digitalization project must be developed and implemented in the company. For the successful completion of a digitalization project, it is helpful to define a person responsible for the project who is already familiar with the implementation of DT. Open communication with employees or customers also plays an important role. In the third phase, the actual implementation of the digitalization project takes place through suitable measures in the company. The final phase monitors the success of the digitalization project. In particular, feedback should be obtained from all stakeholders involved in order to derive the potential for improvement [22].

C. Critical Success Factors

For several decades, practitioners have been dealing with the idea that corporate success is based on specific influencing factors and measures of management. As such, strategies for corporate management have been derived from these influencing factors and measures. As early as 1961, the former McKinsey consultant and later managing director D. R. Daniel developed the theory that management information systems can be used to obtain important information about what he called "success factors" [23], [24]. In practice, success factor research first gained acceptance through the much-cited PIMS study (Profit Impact of Marketing Strategies), which addressed corporate success and its causes in the early 1970s. This was a pioneering study in the field of success factor research. Over the years, other works have also had a significant influence on the domain. For example, Rockart [25] took up the ideas of the initial success factor research and further developed them in his concept of critical success factors using a variety of methods. Rockart [25] conducted intensive interviews with chief executive officers (CEOs) of specific companies to identify success factors. Since 1980, research in the field has changed from specific individual cases to a holistic or industry-specific research of critical success factors [23].

However, the literature defines the term success factors differently. The terms *critical success factors*, *strategic success factors,* and *key factors* are often used as synonyms. In this study, we use the term *critical success factors (CSFs)*. Table 1 shows selected definitions in the literature. The definition by Rockart [25] is the most influential.

TABLE I.
DEFINITIONS OF CSFS

Reference	Definition
[25]	"Critical success factors thus are, for any business, the limited number of areas in which results, if they are satisfactory, will ensure successful competitive performance for the organization. They are the few key areas where 'things must go right' for the business to flourish. If results in these areas are not adequate, the organization's efforts for the period will be less than desired."
[26]	"Key success factors are those variables which management can influence through its decisions that can affect significantly the overall competitive positions of the various firms in an industry."
[27]	"Critical Success Factors (CSFs) are those characteristics, conditions, or variables that when properly sustained, maintained, or managed can have a significant impact on the success of a firm competing in a particular industry."

All authors of the definitions presented in Table I point out that CSFs play a decisive role in the success of the company and the project. They can be seen as areas of action for management to continuously and carefully monitor and contribute to the achievement of the company's goals [25]. However, CSFs vary by company and industry. Therefore, it is important for each company to identify the specific CSFs of their industry and respective project areas.

In the first step of our research project, we identified 25 CSFs of DT (see [18]), which form the basis of this paper. Table II lists the 25 CSFs of DT associated with their respective dimensions. A detailed description of each factor as well as a complete ranking of all 25 CSFs related to both the literature review and the interview study from Step 1 can be requested from the authors.

TABLE II.
CSFS OF DIGITALIZATION PROJECTS (ADAPTED FROM [18])

Dimension	CSFs
Corporate organization	• Corporate culture • Implementation of a digital mindset • Unified digital corporate strategy / vision • Leadership • Top management support • Change management • Digital talent in leadership positions • Qualification
Technology	• Data collection / Big data analysis • Hardware • Software • Unified database in an overall system • Data security
Customer	• Customer centric management model • Omni-channel-management
Project management	• Network effects through open systems / partnerships • Long-term implementation through short intensive sprints • Resources
Value creation	• Networking of the entire value network • Implementation of new KPIs • Cross-functional development teams • Lean thinking / OpEx
Value proposition	• Servitization • Fast prototyping • Scalability

III. STUDY DESIGN

A. Structure of the Online Questionnaire

With our central research question, we aim to gain initial insights into companies' assessments and understandings of CSFs in DT. Therefore, we chose an explorative approach for this study. Accordingly, the study is intended as a *starting point* for more in-depth investigations of the characteristics of the individual CSFs in the further course of our research project. For this reason, we also make no claim of the representativeness of participants in this study.

To design our online questionnaire, we looked at existing CSF study designs (e.g., for ERP system implementation projects) and used them for orientation. In total, our questionnaire was divided in three parts:

Part A comprises 11 questions (Part A.1: four questions, Part A.2: seven questions). In Part A.1, the first two questions address the company's industry sector and number of employees. In the last two questions in part A.1, the participant is asked to state his/her position at the company and the location (federal state) of the company. In Part A.2, first, we asked the participant if he/she agrees with the given definition of DT (see Theoretical Background). The next question discusses the company's attitude regarding DT against the background of the current COVID-19 crisis. We then asked whether the pandemic has favored the attitude towards digitalization projects in certain companies. The next three questions address the participants' assessment of the extent to which their companies have already implemented digitalization projects, in general. For this purpose, the first of the three questions was about the company's status in DT. Here, we asked the respondent to indicate if the company has already embedded DT in its business strategy or whether DT is in the early stages at the company. Secondly, we asked the respondent whether the company has already carried out digitalization projects in individual business areas/departments or is planning to do so. In addition to the status and projects in DT, the next question addressed the estimated degree of DT at the company. The last two questions in part A.2 cover the structure of the IT department and the current digital trends the company is focusing on, such as smart factories or IoT.

Part B comprises seven questions. First, the participant must assess all CSFs regarding the perceived influence on the success of digitalization projects. To this end, we query the 25 CSFs within the dimensions of *business organization*, *technology*, *customer*, *project management*, *value creation*, and *value proposition*. Each dimension represents one matrix question (six questions in total). We used the Likert scale as the psychometric response format. The scale value of the Likert scale could be ultimately calculated as the sum or average score of the respective ratings. We chose a 5-point Likert scale to measure the influence of the CSFs of digitalization projects: *1—No influence*, *2—Little influence*, *3—Medium influence*, *4—High influence*, *5—Very high influence*. After assessing all CSFs, we finally asked the participant to indicate the three CSFs (Top 3 CSFs) they consider most important in DT.

Parts A and B are relevant for addressing our central research question within this paper. The aim of the final **Part C** is to evaluate the implementation of the CSFs in the company. Due to the complexity of the factors, it was, unfortunately, not possible to ask about all CSFs. Therefore, we examined only the three CSFs that were determined in the last question in part B as the three most important CSFs of DT. To prevent the questionnaire from becoming too long, we asked a maximum of three questions for each CSF, so that the total number of questions in Part C did not exceed nine. Hence, the results of Part C will not be part of this paper.

B. Implementation of the Online Questionnaire and Pre-Test

For the implementation of the questionnaire, we used the online survey application LimeSurvey. For a better overview, we displayed all the questions of a question group on one page to reduce the number of clicks needed. We designed the questions of Parts A and B as mandatory questions. The questions of Part C are therefore optional to answer. To ensure the same understanding of the response, we gave all terms/concepts a lay-over possibility, allowing participants to see a given definition.

Before the online survey started, we performed a pre-test. The aim of the pre-test was to check the questionnaire instructions and individual items for comprehensibility and errors. Within the scope of the pretest, seven people from the target group (e.g., managing directors, department heads) went through the questionnaire. Their answers were not included in the final data evaluation. Based on their feedback, we made final changes to the online questionnaire.

C. Data Collection

For the online survey, we invited companies to participate primarily via emails. We used the AMADEUS company database (https://amadeus.bvdinfo.com/) by Bureau van Dijk as the main source for contact information. The query in the AMADEUS database was limited to "active companies," regardless of industry sector, that provided an e-mail address, were headquartered in Germany, and had at least 20 employees. The latter restriction was made due to complexity reduction and, thus, represents a limitation of our study. From the resulting list, 7360 e-mails were randomly sorted and sent to companies in the period from December 1, 2020, to January 31, 2021. In addition, we shared the link to the online survey in various groups on the XING platform (https://www.xing.com/).

After the survey period closed, the questionnaire was at least partially completed 225 times. Of these 225 questionnaires, 101 were completed in full. Before the data analysis was carried out, the 101 fully completed questionnaires were checked for plausibility. During this plausibility check, attention was paid to whether a pattern was discernible in the evaluation of the CSFs with regard to their influence in the success of the digitalization projects, suggesting that the participant had only clicked through the questionnaire at random. In addition, it was checked whether there was a contradiction in the ranking of the CSFs with its evaluation. We, therefore, needed to exclude four data sets, which meant that 97 data sets could be taken into account for the evaluation of results presented in the following chapter.

IV. Selected Results

A. General Participants' Characteristics

First, we asked the 97 participants about the general characteristics of their companies, which included location, industry affiliation, number of employees, and position of the participants (**Part A.1**).

Companies from all German federal states took part. Most of the participants came from companies in Berlin (n=13) and Lower Saxony (n=10); the locations of the other participants are balanced across the other federal states. Since we make no claim to representativeness, we have not further divided the results according to the company shares per federal state.

Most of the companies (n=22) belong to the manufacturing industry/production of goods. The subsequent dominant sector allocation falls to the provision of economic services (n=18) and education and training (n=11). The remaining companies are distributed roughly equally among the other industry sectors. The aggregation of the individual sectors to the secondary sector (industrial production) or tertiary sector (service enterprises in the broader sense) shows that most companies belong to the service sector (n=65), and the remaining businesses are industrial enterprises (n=32). According to the indicated number of employees, most of the companies (n=70) are SMEs (i.e., companies with up to 249 employees). Large companies are in the minority in our sample (n=27). Approximately half of the participants (n=50) belong to top management or executive management, whereby 19% of the participants (n=18) hold the position of professionals within a specific department. Furthermore, 12 department managers and 13 project managers participated in the survey. Four participants held other positions (e.g., business development, digital officer).

B. Digital Transformation within the Companies

Following the general question regarding company specifics, we asked seven questions with a specific focus on the characteristics of DT (**Part A.2**).

First, participants were asked to evaluate **(1)** a presented definition of DT (see Theoretical Background). Almost two-thirds of the participants (n=63) fully agreed with the given definition. One-third of the respondents (n=32) agreed at least partially. Reasons for partial agreement with the DT definition vary widely. For example, it was noted that each company must overcome individual challenges in the context of DT, and that the definition can, therefore, only be regarded as a rough guide. Furthermore, participants put into perspective that new business models and strategies at existing companies are not necessary for the success of DT.

The companies then assessed to what extent their **(2)** attitude towards DT has changed due to the COVID-19 crisis. The majority of companies (n=58) indicated that their attitude toward DT has not changed as a result of the current COVID-19 crisis, since they had already perceived DT as an important issue. This indicates that many companies had already addressed DT in their strategies or are currently doing so. One-third of the companies (n=32) perceived DT as more important than before due to the COVID-19 crisis. In turn, five companies indicated that their attitudes toward DT have not changed because of the COVID-19 crisis, in that DT does not play an important role in their companies.

Furthermore, the **(3)** DT status of the company was of interest: DT was already an integral part of the business strategy in almost half of the companies surveyed (n=47). In 40% of the companies, there was no overarching corporate strategy for DT, but they had already started or implemented single digitalization projects. Ten companies are currently in the planning phase in digitalization projects, and only one company indicated that it has not yet addressed the issue of DT at all.

The companies were then asked about **(4)** digitalization projects conducted or planned along key business functions (logistics, production, human resources, purchasing, sales, marketing, accounting/controlling, service, other). In human resources (n=61), marketing (n=54) and accounting/controlling (n=56) functions, most companies have already conducted digitalization projects. One in three companies—cumulatively viewed for all functions—is currently conducting or has already completed digitalization projects.

When asked about the **(5)** degree of DT at the company, participants were asked to indicate the extent to which they consider their company to be digitalized on a scale from "0" (not digitalized at all) to "10" (fully digitalized). Most companies (n=76) rated their company's level of DT as 5 to 8. Seventeen companies rated themselves with categories of 0 to 4. The remaining companies assigned themselves scores of 9 or 10.

This was followed by the question on **(6)** the structure of the IT department (multiple answers were allowed). Most companies (n=37 each) stated that they either have a central and, therefore, "classic" IT department and/or employ an external IT service provider. Approximately 20% do not have their own IT department, and 15% employ IT experts directly in individual departments. In nine companies, the IT department is *bimodal*, which allows the companies to accelerate and drive their digitalization projects in a separate infrastructure. Two companies also indicated that decentralized IT departments exist per functional area.

In the final question of Part A.2, we asked for **(7)** the DT trend topics the companies have already addressed. The topic that most companies (n=66) have already addressed or are currently focusing on is cloud technologies. While many companies (n=34) are also focusing on big data, some are also dealing with trends like additive manufacturing processes, IoT, cyber-physical systems, and smart factory. The trends a company chooses to address often also depend on the industry sector. For example, cyber-physical systems or smart factory play a role more often in the manufacturing sector and less frequently in service companies. Some companies also listed additional trends, i.e., artificial intelligence (AI), telematics infrastructure, and hybrid commerce.

C. Assessment of CSFs for Digitalization Projects

The core of our survey consisted of assessing all identified CSFs (see [18]) in terms of their influence on the success of digitalization projects (**Part B**). First, companies rated their importance using the 5-point Likert scale (*1—No influence, 2—Little influence, 3—Medium influence, 4—High influence, to 5—Very high influence*). Second, participants chose the three CSFs they considered the most important in terms of the success of digitalization projects (Top 3 CSFs). The respective rankings are shown in Table III. The following results are referring to the left-hand column of Table III. The results of the Top 3 CSFs (right-hand column of Table III) are taken up later in the Discussion section.

The dimension *Corporate organization* is the largest and comprises eight CSFs. The entire dimension seems to have a high to very high impact (on average, rated with a 4.14), as the participants mostly rated the pertinent CSFs with a four or five:

- About nine out of ten companies rated the CSF of *Corporate culture* as very important for digitalization projects.
- Most companies rated the CSFs *Implementation of a digital mindset* and *Unified digital corporate strategy/vision* as high (n=46; n=40) to very high (n=37; n=38).
- About eight out of ten companies believe that the CSF *Leadership* has a high (n=32) or very high (n=44) impact for digitalization projects.
- For the CSF *Top management support*, over 50% of the respondents (n=52) indicated that this factor has a very high influence in DT project implementation.
- For *Change management* and *Digital talent in leadership positions*, the percentage of companies rating the influence as only moderate is higher (n=20; n=16,) than for the other CSFs in this dimension. However, even for these two CSFs, companies rated their influence as high (n=40; n=43) or very high (n=31; n=24).
- The final CSF in this dimension, *Qualification*, is also rated as having a high (n=49) to very high (n=32) influence with respect to the success of digitalization projects.

TABLE III.
COMPARISON OF THE TWO DIFFERENT RANKINGS OF CSFS FROM QUESTIONNAIRE PART B

Ranking of CSFs based on average score using the 5-point Likert scale		**Ranking of CSFs based on the indication of the Top 3 CSFs**	
Critical success factor	**Rank**	**Critical success factor**	**Rank**
Data security	1	Corporate culture	1
Software	2	Unified digital corporate strategy / vision	2
Top management support	3	Implementation of a digital mindset	3
Unified database in an overall system	4	Top management support	4
Corporate culture	5	Qualification	5
Implementation of a digital mindset	6	Leadership	6
Unified digital corporate strategy / vision	7	Unified database in an overall system	7
Leadership	8	Software	8
Qualification	9	Change management	9
Resources	10	Digital talent in leadership positions	9
Change management	11	Data security	11
Networking of the entire value network	12	Resources	12
Digital talent in leadership positions	13	Data collection / Big data analysis	13
Cross-functional development teams	14	Customer centric management model	14
Hardware	15	Cross-functional development teams	14
Customer centric management model	16	Servitization	14
Long-term implementation through short intensive sprints	17	Networking of the entire value network	14
Scalability	18	Hardware	18
Network effects through open systems / partnerships	19	Long-term implementation through short intensive sprints	19
Lean thinking / OpEx	20	Implementation of new KPIs	20
Data collection / Big data analysis	21	Omni-channel-management	20
Omni-channel-management	22	Fast prototyping	20
Servitization	23	Lean thinking / OpEx	23
Fast prototyping	24	Network effects through open systems / partnerships	23
Implementation of new KPIs	25	Scalability	23

The dimension *Technology* is the second largest dimension and includes five CSFs. This dimension is also assigned a high to very high influence, as the individual CSFs were predominantly rated as a four or five. On average, companies rated all CSFs in this dimension with 4.11:

- The CSF *Data security* stands out in having the highest influence on project success in DT: 23 participants perceive a high influence on digitalization projects. Two thirds of the companies (n=64) stated that the influence of this factor is very high.
- For the two CSFs *Software* and *Unified database in an overall system*, the influence on the successful implementation of digitalization projects is mainly rated as high (n=40; n=39) to very high (n=50; n=44).
- The assessment was not so clear-cut for the last two CSFs *Data collection / Big data analysis* and *Hardware*. In both cases, participants agreed to a high (n=34; n=41) or very high impact (n=18; n=22). Compared to the other three CSFs in this dimension, respondents also indicated that these two CSFs each had a rather medium influence (n=24 and n=21). In addition, about one in ten respondents (n=11 and n=12) rated the influence as low in each case.

The dimension *Customer* covers the two CSFs *Customer centric management model* and *Omni-channel-management*. On average, respondents in this dimension rated the impact of the CSFs on project success in DT only with a 3.54: About half of the respondents each rated the influence of the two CSFs as high to very high. Just under one-fifth of the companies (n=18; n=21) rated the influence as medium. Compared to the CSFs of the first two dimensions considered so far, some participants stated that these CSFs have no influence on the success of digitalization projects. This may be due, for example, to the fact that these two CSFs are somewhat more specific for individual industry sectors and many respondents may not be able to assess this for their company.

The three CSFs *Network effects through open systems / partnerships*, *Long-term implementation through short intensive sprints*, and *Resources* belong to the dimension *Project management*. On average, this dimension is rated with a 3.74. This is slightly above the score for the dimension *Customer* (3.54) but below the dominant ones (*Corporate Organization*: 4.14; *Technology*: 4.11). All three CSFs of this dimension were assigned a high influence on the success of digitalization projects by over 40% of the companies (n=40, n=45 and n=43). Almost one-third of respondents (n=31) even rated the influence of *Resources* as very high. In contrast, for the other two CSFs, a quarter of the companies (n=24 and n=26) think that the influence on project success in DT is rather moderate. Furthermore, 15% of the participants (n=12 and n=13) believe that the CSF *Network effects through open systems / partnerships* has little to no influence on the success of the digitalization projects.

The dimension *Value creation* consists of four CSFs. On average, there is a rating of 3.59 in this dimension. The influence of the two CSFs *Networking of the entire value network* and *Cross-functional development teams* was rated higher than for the other two CSFs. More than 40% of the companies (n=39; n=43) indicated that the two CSFs mentioned had a high influence on project success in DT, and several companies (n=23; n=19) even rated this as very high. However, about 22% of the participants (n=22 and n=21) are of the opinion that the two CSFs have only a medium influence on success. With regard to the other two CSFs, *Implementation of new KPIs* and *Lean thinking / OpEx*, most companies (n=35; n=34) stated that the influence here is neutral.

The final dimension *Value proposition* includes the three CSFs *Servitization*, *Fast prototyping*, and *Scalability*. On average, participants rated this dimension the lowest of all dimensions with a 3.38. When evaluating the CSF *Servitization*, 27 of the respondents stated that its influence on the success of the digitalization projects is medium, 21 of the respondents estimated it to be high, and 10 companies very high. For *Fast prototyping* and *Scalability*, approximately 30% of respondents (n=29; n=30) believe that their influence on project success is high. The percentage of respondents who find their influence to be neutral is one-fifth (n=20) and one-quarter (n=23), respectively.

V. Discussion

Regardless of the size of the company, projects within the scope of DT are complex and extensive undertakings, which sometimes lead to strong interventions in the company's processes and daily business. A structured approach to the implementation of digitalization projects can prove highly useful. Therefore, it can be helpful for companies to use CSFs as a guide for the specific implementation of digitalization projects. According to the assessments of the companies surveyed, organizational factors, in particular, play a decisive role in DT. *Corporate culture* was rated as the most important CSF for digitalization projects (with a view to the Top 3 CSFs, right-hand column in Table III). DT gives rise to new business models, thus companies must adapt and improve business processes for these new circumstances. As these changes often collide with the already existing corporate culture, it is particularly important that companies invest time and resources to create a digital corporate culture. The associated, necessary changes should be openly communicated and, above all, implemented together with the employees (internal co-creation). The newly created corporate culture should have flatter hierarchies and be data-based, data-driven, agile, risk-aware, and creative [28]. However, *Top management support* is also necessary for successful implementation. Without certain commitment and project understanding, most digitalization projects will fail. Since the top management is responsible for the digital transformation strategy, it must also be actively involved in the digital transformation process. Managers should define appropriate goals for DT and harmonize them with the rest of the corporate goals. Regarding DT, it is necessary that the future positioning of a company is anchored in a *Unified digital corporate strategy / vision*. For a successful implementation of digitalization projects, companies must define corresponding goals, determine expected developments, and derive resulting measures. Due to a strong dependency of corporate strategy and corporate culture, it is important that both are aligned [16], [29]. A particularly helpful exposition for successfully aligning is to implement a digital mindset in the company.

Setting the right course at the organizational and strategic level is one thing—the effective and efficient implementation of concrete projects for DT is another. Since digitalization projects—in their dominant nature as innovation projects (new-to-the-firm or even new-to-the-market)—can have different focuses, not all factors are of the same relevance for every company or every project. Noticing this, we set the identified success factors in relation to the different dimensions of DT. This resulted in a comprehensive model as a starting point for digitalization projects—see Fig. 1. The ranks of the Top 10 CSFs shown in this figure refer to the left-hand column in Table III.

When the survey results are integrated into our model (see Fig. 1), the abovementioned discussion becomes clear once again: Most of the Top 10 CSFs are assigned to the dimension *Digital corporate organization*. The other Top 10 CSFs (except for the CSF *Resources*) belong to the dimension *Digital technology*. This can be interpreted that the two dimensions are particularly fundamental and flanking areas, which are, therefore, rated as the most important. The *Digital corporate organization* permeates the other dimensions of DT and influences all transformation tasks. The technological basis is, in turn, a component of all developments in the other dimensions.

Differences emerge in the results when the data provided by SMEs (n=70) and large companies (n=27) are considered separately:

- Top 3 CSFs of SMEs: *Data security*, *Software*, and *Top management support*.
- Top 3 CSFs of large enterprises: *Data security*, *Corporate culture*, and *Implementation of a digital mindset*.

Looking at the Top 10 CSFs without considering company size (see Table III, left column), the difference between large companies and SMEs is shown, in that both SMEs and large companies consider the dimensions of both *Digital corporate organization* and *Digital technology* to be of particular importance. In the case of SMEs, however, technological CSFs rank even higher, whereas large companies give more importance to organizational CSFs. One possible explanation for this difference is that large companies have more resources (human and financial) to create the technological basis for DT—and have already done so to a much greater extent than SMEs.

The distinction between SMEs and large enterprises can shed new light on CSFs, since these have to be interpreted through the background of company specifics in structure and processes (e.g., flatter structures, familiarity, scarcity of resources in SMEs). Since organizations are social systems that consist of complex interactions between individuals and groups, *Corporate culture (*as a CSF of the dimension *Digital corporate organization*) is, therefore, incorporated. In this way, shared convictions and attitudes exist in groups, which influence the perception, reactions to changes, and, consequently, the occurrence of resistance/barriers [30]. *Corporate culture*, thus, has an influence on perceptions, attitudes, and behaviors [31], which, in turn, influence the success of digitalization projects [32].

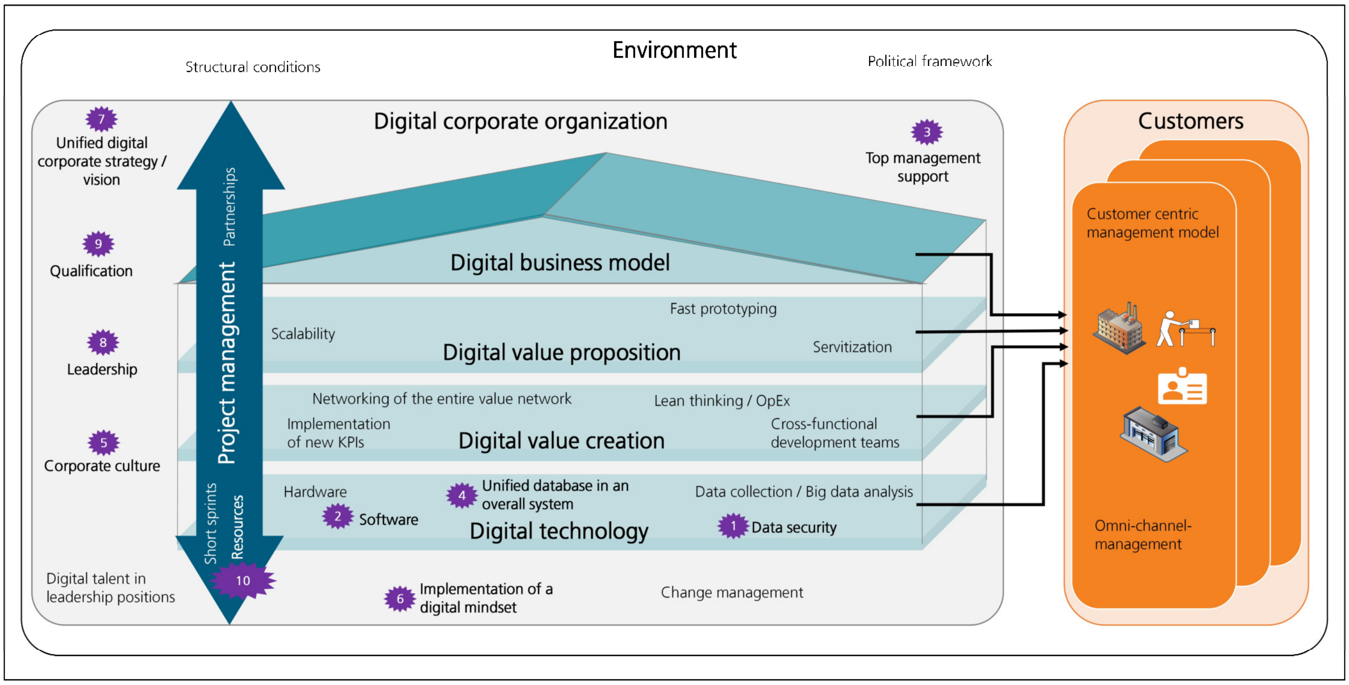


Fig. 1. Model of Digital Transformation – Integration of Dimensions and Critical Success Factors

VI. Conclusion

The results of our study make a significant contribution to the CSF research focusing on digitalization projects. Future research activities in this topic area can build on the insights gained from our study. For example, individual factors, such as the Top 3 CSFs *Corporate culture*, *Unified digital corporate strategy / vision*, and *Top management support*, could be investigated in more detail to derive recommendations for action for the best possible implementation of the CSFs in the company. Furthermore, qualitative and quantitative studies (with claim to representativity) can be conducted in individual industry sectors and with a more specific consideration of company sizes to further specify the importance of CSFs for digitalization projects in this regard. Another starting point for future research could be to analyze CSFs with reference to the different types of digitalization projects, such as logistics or human resources, to highlight any differences. Furthermore, it should be investigated what makes the implementation of individual CSFs in companies more difficult and how these obstacles can be minimized.

The need for a more detailed and diversified view of different CSFs becomes even more evident when considering the potential impact of the COVID-19 crisis on DT. The crisis brought DT into sharper focus, especially for companies that had not previously addressed DT in such detail. This is also illustrated by the answers to the question focusing on the influence of the COVID-19 crisis. Even though nearly 60% of the companies had seen DT as important before the pandemic, an additional 33% now see DT as more important than before. In conclusion, this shows the importance of focusing strongly on DT in research and deriving concrete practice-oriented recommendations for action and assistance for companies in shaping DT. By discussing CSFs within the context of the COVID-19 crisis, different questions on short-term and long-term time horizons are implied, e.g., what changes were organizations able to implement ad hoc, what are the lessons learned, which changes will remain after the COVID-19 crisis? At the interface of digitalization projects, the call for new work imperatives came up. However, since the advancement/adaption strategies of large companies are often clearer than the respective coping mechanisms of SMEs, we are currently working on a study that focuses on CSFs for improved data management and data analysis within SMEs (as an exemplary digitalization project) in times of the COVID-19 crisis.

References

[1] R. G. Fichman, B. L. Dos Santos, and Z. (Eric) Zheng, “Digital Innovation as a Fundamental and Powerful Concept in the Information Systems Curriculum,” *MIS Quarterly*, vol. 38, no. 2, pp. 329–343, 2014, doi: 10.25300/MISQ/2014/38.2.01.

[2] C. Leyh, T. Schäffer, K. Bley, and S. Forstenhäusler, “Assessing the IT and Software Landscapes of Industry 4.0-Enterprises: The Maturity Model SIMMI 4.0,” in *Information Technology for Management: New Ideas and Real Solutions*, Lecture Notes in Business Information Processing, LNBIP, Vol. 277, E. Ziemba, Ed. Cham: Springer, 2017, pp. 103–119. doi: 10.1007/978-3-319-53076-5_6.

[3] M. Pagani, “Digital Business Strategy and Value Creation: Framing the Dynamic Cycle of Control Points,” *MIS Quarterly*, vol. 37, no. 2, pp. 617–632, 2013, doi: 10.25300/MISQ/2013/37.2.13.

[4] C. Leyh, K. Bley, and M. Ott, “Chancen und Risiken der Digitalisierung – Befragungen ausgewählter KMU,” in *Arbeit 4.0 – Digitalisierung, IT und Arbeit*, J. Hofmann, Ed. Wiesbaden: Springer, 2018, pp. 29–51. doi: 10.1007/978-3-658-21359-6_3.

[5] S. Mathrani, A. Mathrani, and D. Viehland, “Using enterprise systems to realize digital business strategies,” *Journal of Enterprise Information Management*, vol. 26, no. 4, pp. 363–386, 2013, doi: 10.1108/JEIM-01-2012-0003.

[6] K. Bley, C. Leyh, and T. Schäffer, “Digitization of German Enterprises in the Production Sector - Do they know how ‘digitized’ they are?,” in *Proceedings of the 22nd Americas Conference on Information Systems (AMCIS 2016)*, 2016.

[7] C. Helmenstein, M. Zalesak, J. El-Rayes, and P. Krabb, “Raise the Curve: Mit Digitalisierung zu mehr Resilienz und Wachstum,” Accenture and Industriellenvereinigung, 2020.

[8] A. Berg, “Digitalisierung der Wirtschaft – Auswirkungen der Corona-Pandemie,” Berlin: bitkom, 2020.

[9] K.-H. Streibich and J. Winter, “Resiliente Vorreiter aus Wirtschaft und Gesellschaft,” Munich: acatech — Deutsche Akademie der Technikwissenschaften, 2020.

[10] Deutsche Telekom AG, “Der digitale Status quo des deutschen Mittelstands - Digitalisierungsindex Mittelstand 2020/2021,” techconsult GmbH and Deutsche Telekom AG, 2020.

[11] A. Jones, J. Robinson, B. O’Toole, and D. Webb, “Implementing a bespoke supply chain management system to deliver tangible benefits,” *The International Journal of Advanced Manufacturing Technology*, vol. 30, no. 9–10, pp. 927–937, 2006, doi: 10.1007/s00170-005-0065-2.

[12] R. Hentschel, C. Leyh, and T. Baumhauer, “Critical Success Factors for the Implementation and Adoption of Cloud Services in SMEs,” in *Proceedings of the 52nd Hawaii International Conference on System Sciences (HICSS 2019)*, 2019.

[13] P. Achanga, E. Shehab, R. Roy, and G. Nelder, “Critical success factors for lean implementation within SMEs,” *Journal of Manufacturing Technology Management*, vol. 17, no. 4, pp. 460–471, 2006, doi: 10.1108/17410380610662889.

[14] C. Leyh and J. Thomschke, “Critical Success Factors for Implementing Supply Chain Management Systems – The Perspective of Selected German Enterprises,” in *Proceedings of the 2015 Federated Conference on Computer Science and Information Systems (FedCSIS 2015)*, 2015, pp. 1403–1413. doi: 10.15439/2015F245.

[15] J. M. Denolf, J. H. Trienekens, P. M. (Nel) Wognum, J. G. A. J. van der Vorst, and S. W. F. (Onno) Omta, “Towards a framework of critical success factors for implementing supply chain information systems,” *Computers in Industry*, vol. 68, pp. 16–26, 2015, doi: 10.1016/j.compind.2014.12.012.

[16] F. Holotiuk and D. Beimborn, “Critical Success Factors of Digital Business Strategy,” in *Proceedings der 13. Internationalen Tagung Wirtschaftsinformatik (WI 2017)*, 2017.

[17] C. Leyh and L. Crenze, “ERP System Implementations vs. IT Projects: Comparison of Critical Success Factors,” in *Enterprise Information Systems of the Future*, Lecture Notes in Business Information Processing, LNBIP, Vol. 139, G. Poels, Ed. Berlin, Heidelberg: Springer, 2013, pp. 223–233. doi: 10.1007/978-3-642-36611-6_20.

[18] C. Leyh and N. Meischner, “Erfolgsfaktoren von Digitalisierungsprojekten - Einflussfaktoren auf Projekte zur Digitalen Transformation von Unternehmen,” *ERP Management*, vol. 2/2018, pp. 35–38, 2018, doi: 10.30844/ERP18-2_35-38.

[19] R. Sauer, M. Dopfer, J. Schmeiss, and O. Gassmann, “Geschäftsmodell als Gral der Digitalisierung,” in *Digitale Transformation im Unternehmen gestalten: Geschäftsmodelle, Erfolgsfaktoren, Handlungsanweisungen, Fallstudien*, O. Gassmann and P. Sutter, Eds. Munich: Hanser, 2016, pp. 15–27.

[20] E. Wallmüller, *Praxiswissen Digitale Transformation: Den Wandel verstehen, Lösungen entwickeln, Wertschöpfung steigern*. Munich: Hanser, 2017. doi: 10.3139/9783446452732.

[21] D. Barghop, E. Deekeling, and D. Schweer, “Herausforderung Disruption: Konsequenzen und Erfolgsfaktoren für die Kommunikation,” in *Kommunikation in der digitalen Transformation*, E. Deekeling and D. Barghop, Eds. Wiesbaden: Springer, 2017, pp. 5–19. doi: 10.1007/978-3-658-17630-3_2.

[22] C. Falkenreck, *Digitalisierungsprojekte erfolgreich planen und steuern: Kunden und Mitarbeiter für die digitale Transformation begeistern*. Wiesbaden: Springer, 2019. doi: 10.1007/978-3-658-24890-1.

[23] A. Nicolai and A. Kieser, “Trotz eklatanter Erfolgslosigkeit: Die Erfolgsfaktorenforschung weiter auf Erfolgskurs,” *Die Betriebswirtschaft*, vol. 62, no. 6, pp. 579–596, 2002.

[24] D. R. Daniel, “Management Information Crisis,” *Harvard Business Review*, vol. 39, no. 5, pp. 111–121, 1961.

[25] J. F. Rockart, “Chief executives define their own data needs,” *Harvard Business Review*, vol. 57, no. 2, pp. 81–93, 1979.

[26] C. W. Hofer and D. Schendel, *Strategy formulation : analytical concepts*. St. Paul/Minnesota: West Publishing, 1978.

[27] J. K. Leidecker and A. V. Bruno, “Identifying and using critical success factors,” Long Range Planning, vol. 17, no. 1, pp. 23–32, 1984, doi: 10.1016/0024-6301(84)90163-8.

[28] G. Wokurka, Y. Banschbach, D. Houlder, and R. Jolly, “Digital Culture: Why Strategy and Culture Should Eat Breakfast Together,” in Shaping the Digital Enterprise, G. Oswald and M. Kleinemeier, Eds. Cham: Springer, 2017, pp. 109–120. doi: 10.1007/978-3-319-40967-2_5.

[29] J. vom Brocke, M. Fay, T. Schmiedel, M. Petry, F. Krause, and T. Teinzer, “A Journey of Digital Innovation and Transformation: The Case of Hilti,” in Shaping the Digital Enterprise, G. Oswald and M. Kleinemeier, Eds. Cham: Springer, 2017, pp. 237–251. doi: 10.1007/978-3-319-40967-2_12.

[30] B. J. Weiner, “A theory of organizational readiness for change,” Implementation Science, vol. 4, no. 67, 2009, doi: 10.1186/1748-908-4-67.

[31] Q. Hu, T. Dinev, P. Hart, and D. Cooke, “Managing Employee Compliance with Information Security Policies: The Critical Role of Top Management and Organizational Culture,” Decision Sciences, vol. 43, no. 4, pp. 615–660, 2012, doi: 10.1111/j.1540-5915.2012.00361.x.

[32] S. Ries, “Veränderungen in kleinen und mittelständischen Unternehmen: Innerbetrieblichen Widerstand überwinden, Unterstützung von Veränderung herbeiführen und organisationale Veränderungsbereitschaft leben,” Leipzig: Fraunhofer Center for International Management and Knowledge Economy IMW, 2021.

DOI: 10.15439/2021F64
ISSN 2300-5963 ACSIS, Vol. 25

The Impact of Digital Technologies on How Companies Work: Results from an Interview Study

Christian Leyh
Fraunhofer Center for International Management and Knowledge Economy IMW
Neumarkt 9-19, 04109 Leipzig, Germany
Email: christian.leyh@imw.fraunhofer.de

Paul Becke
Master Student at Leipzig University
Chair of Innovation Management and Innovation Economics
Grimmaische Str. 12,
04109 Leipzig, Germany

Milan Pentrack, Bastien Bodenstein
Fraunhofer Center for International Management and Knowledge Economy IMW
Neumarkt 9-19, 04109 Leipzig, Germany
Email: milan.pentrack@imw.fraunhofer.de, bastien.bodenstein@imw.fraunhofer.de

Abstract—**The increasing digitalization of business and society has prompted drastic changes within enterprises and confronted them with enormous challenges. In our exploratory interview study, we examined the impact of digital technologies on how employees work in companies as well as the specific opportunities and challenges that small and medium-sized enterprises (SMEs) face as a result. On the whole, interviews revealed that digitalization has already triggered an array of changes in how employees work. Even so, the extent of each change and the perception of technological trends overall have varied among both employees and companies depending on their context. In response to those changes, the SMEs interviewed have applied a wide range of tools and strategies that have allowed them to exploit the opportunities offered by digital technologies and overcome the associated challenges.**

I. Introduction

SOCIETY as a whole is undergoing a rapidly evolving digital transformation, one in which governmental institutions, households, companies, and their interactions are changing due to the increasing spread of digital technologies. As a result, it has never been more important, especially for companies, to be able to rely on capabilities enabled by information technology (IT) or on a deep understanding of IT in general and digital innovation in particular. As part of the evolution of technology, digitalization provides numerous unprecedented opportunities to support and even renew business processes. In turn, those advanced technological opportunities, particularly ones that merge the physical and digital worlds, have brought about new paradigm shifts that affect all industry sectors. By extension, stable, prevailing dynamics in everyday business show that constant changes and adjustments, including digitalization, will not be the exception but the rule in economies of the future. The consequences of that development and the question of whether it should be viewed as positive or negative are omnipresent. Perhaps most saliently, formerly analog activities—reading a newspaper, for example, or buying a physical product—have acquired digital twin processes that can be performed on mobile devices at any place and at any time [1]–[6]. That trend was jolted forward by the COVID-19 pandemic, which has further disrupted how businesses operate and how traditional services are delivered. As a consequence, the digital expectations of consumers and B2B customers have reached new heights. In response, some companies have rapidly digitalized their interactions with customers and the supply chain as well as their internal operations, sometimes even by 3 to 4 years relative to their competitors [7], meaning that those competitors now face overwhelming lags in their digital capabilities.

No matter the pace of digitalization, continuous interaction with technology in both professional and personal settings has become more standard than ever before. In that environment, studying human behavior in organizations without considering the influence of IT is short-sighted [8]. For that reason, Daugherty and Carrell-Billiard [9] have used the term “human+” to describe the workforce of the digital age, whose members not only possess their pre-existing talents and knowledge but also have new, expanding sets of skills acquired by simply using digital technologies.

From that perspective, one strand of literature addresses the possibilities of improving the quality of work and of private life by employing modern technologies and compensating for their negative effects [10]. At the other extreme, another strand focuses on those negative effects and attempts to assess the consequences faced by the human workforce in particular [11]. Between those strands, many papers describe the effects of digitalization or the digital transformation of companies, usually with reference to case studies due to the subject’s topicality [12]. In both scientific studies and reports by management consultancies and market research companies, however, the effects for employees have received less attention than those for the economy.

Given the subject’s topicality and the rise of literature addressing it, having too few qualitative studies and comprehensive literature reviews may have contributed to an inconsistent definitional framework [12]. Meanwhile, only a handful of reports issued by government and private-sector interest groups, as well as only a small proportion of scientific publications, are dedicated to digitalization’s effects on small and medium-sized enterprises (SMEs). Although SMEs bear great economic significance, especially in Germany, digitalization's effects in such enterprises have hardly been examined.

In contribution to the current state of research, this paper addresses the field of effects digital technologies have on how companies work. Framed by an overview of their general effects on employees, the paper specifically explores the extent to which digital technologies already play an important role in SMEs, whether employees' work practices have changed as a result, and, if so, then how. To support our argument, we conducted an interview study with SME practitioners that followed an exploratory research approach, in which we sought to investigate and identify possibilities and challenges for the work environments of SMEs as a result of using digital technologies and of digitalization in general.

To appropriately situate and present our study and its results, the paper is structured as follows. In Section II, we provide a brief theoretical background on digitalization as well as its effects on employees. Next, in Section III, we describe the methods of data collection used in our interview study. After that, in Section IV, the primary part of the paper, we provide selected results from the interview study and, in Section V, discuss those results. In Section VI, we reflect on what the results imply for practice in the form of recommendations for action, and in Section VII, we conclude the paper with an outlook for further research.

II. Theoretical Background

A. Digitalization and digital transformation

Despite steady growth in scientific literature on digitalization, such research has often focused exclusively on individual technologies or industries. Even then, company-specific case studies represent only a small fraction of that overall development [13]. In effect, the state of knowledge in that area of research is highly fragmented and not always based on consistent assumptions.

That effect is already evident in the two different meanings of the terms *digitization* and *digitalization*, which are sometimes used synonymously in science and business. Whereas *digitization* describes the pure transition from analog to digital data or services, *digitalization* is used to emphasize changes in processes, value chains, and business models, among other things, that go beyond the mere digitization of existing processes and structures—that is, that create added value [14], [15]. In this paper, our focus is on the term *digitalization*, the goal of which is the digital transformation, or the digital change, of organizations—in our case, companies. Guided by that focus, our research targeted the added value resulting from interactions between digital technologies implemented at companies and the employees affected by them.

B. Impact of digitalization on employees

Bonin, Gregory, and Zierahn [16], after replicating Frey and Osborne's [11] study on the likelihood of the automation of U.S. professions in the German context, have concluded that technologies do not necessarily displace jobs as long as employees continuously adapt their skills to new circumstances, learn to use new technologies, and focus on excelling in activities that are difficult to automate. Added to that, Autor [17] has argued that technology rarely replaces entire jobs but often complements human labor by automating individual processes and may even create additional jobs under certain circumstances. For example, individuals in managerial, professional, and technical professions involving abstract tasks can particularly benefit by being able to analyze information more easily, more cheaply, and on a larger scale by using digital technologies, as well as by spending more time on the value-adding activities of interpreting and applying the information [17]. Autor [17] has therefore identified the greatest potential for change and automation in the routine work of knowledge workers, especially in office jobs, and manufacturing workers, whose simple calculation, data collection, transmission and storage, and precise standardized production processes are ripe for digitalization. For that reason, the work environment of those occupational groups and related industries was the focus of our study.

III. Research Methodology

A. Research design

The aim of our interview study was not to generate a new theory on the basis of interpretative generalization, for the data collected for that purpose would have to be considered in the context of the respective organization [18]. Neither was the aim to explore digital transformation in general, given that a comprehensive and growing body of literature on that topic already exists. On the contrary, our interview study, following an exploratory research design, sought to uncover seldom-observed problems created by digitalization's impacts on employees in light of the experiences of practitioners themselves. To that end, we decided to employ a qualitative approach that considers the personal perceptions, motives, background, and experience of experts in a more comprehensive, detailed way than possible with any quantitative approach [19]. More specifically, our approach can be regarded as systematizing expert interviews of an explorative character and that foreground the data's thematic comparability [20]. To ensure such comparability of the interview results, the interviews were conducted using a semistructured interview guide.

B. Selection of experts

Relevant experts were executives and managing directors of SMEs who have both insights into the technological infrastructure of their companies and can assess that infrastructure's impact on their personal work methods and those of their colleagues. To be able to compare the experts' statements, the search was limited to two specific industry sectors whose adoption of digital technologies shows extraordinary potential [17]: B2B manufacturing and banking. Accordingly, 90 SMEs were sent a cover letter regarding the study via email, and we were ultimately able to conduct interviews with 14 experts. In both sectors, the same number of experts was interviewed. Table I in Appendix B provides an overview of the interviewed companies and the positions of the expert interviewees.

C. Data collection and analysis

The interview guide consisted of four blocks of questions, each with four to six primary questions, along with situational follow-up and sub-questions to be asked as needed: (1) general questions, (2) flexibility via digitalization, (3) data analysis, and (4) automation. The interview guide with each block's primary questions appears in Appendix A.

All 14 expert interviews were conducted over the phone between February 25 and April 16, 2020, and proceeded according to the interview guide, which had previously been sent to the experts via email for preparation. The interviews, varying in length from approximately 35 minutes to 2 hours, were recorded with the consent of the interviewees and fully transcribed and anonymized, with dialects and grammatical errors in the conversations partly transcribed into standard language to make the content more comprehensible [21]. Before analysis, the experts received the opportunity to make further requests for changes, and approval was obtained for the selected degree of anonymization.

The subsequent, computer-assisted coding of relevant text passages was conducted using the data analysis software MAXQDA 12. The coding system used was based, on the one hand, on the structure of the interview guide in order to enable the clearest, most systematic coding possible. On the other, care was taken to ensure that the theses derived from the literature search could later be discussed in a differentiated manner for each industry sector. The coding system was tested in the first three interviews, minor changes were made (e.g., the code "digital departments/responsible persons" was inserted in the area "digital transformation" because new departments had been created in all companies in recent years), and then retained for all subsequent interviews.

IV. Interview Results

Building on the preliminary literature review, the expert interviews confirmed numerous theoretical sources of potential and challenges posed by digital technologies and trends for SMEs. Overall, the interview study's results supported the assumption that SMEs in the industry sectors under consideration (i.e., manufacturing in the B2B sector and the banking industry) face additional challenges relative to larger companies. The ways how SMEs work have also been decisively influenced by digitalization, and it is evident general deductions from research findings only reflect part of the reality in companies and should therefore be supplemented with more specific findings from practice in order to appropriately assess the impact of digital technologies. The interviews revealed that perceptions of digital change and associated internal company developments are industry-, company- and person-specific, which is why the sources of potential and challenges identified always need to be considered in their respective contexts in order for the reasons for the assessments made to be understood. For example, the reduction of personnel can be seen as both an opportunity and a risk depending on the context, and the assessment can differ not only within an industry sector but also within a company.

A. Differences between the industry sectors

As expected, differences emerged between the two industry sectors in terms of the reasons for digitalization and its various forms. The SME banks examined view themselves as service providers, such that the demand of their customers for digital offerings primarily drives their digital transformations at the process and product levels. The study did not reveal any company in the banking sector that continues to exist in the market without a minimum level of digital processes and offerings. In manufacturing in the B2B sector, by contrast, the products and development services have not necessarily changed. Instead, new and complementary digital services are emerging in isolated cases, and some companies, including the manufacturer of control solutions M, exhibit a trend of offering complete solutions instead of pure products, although most of the changes described had occurred at the process level. In that case, digitalization has been driven by the overall market and the companies themselves in order to remain competitive. However, examples such as the measurement technology manufacturer L show that even manufacturers with a comparatively low level of digitalization can currently hold their position against the competition. Despite those differences, both banks and manufacturing companies are increasingly developing into technology companies in the course of the third and fourth industrial revolutions (Experts C and D). No expert explicitly assumed that their company could survive in the market in the long term without its own know-how in digitalization and the use of digital technologies. Other digital technologies described by the interviewees imply that SMEs have already begun working with innovative technologies (e.g., distributed ledger technologies or cryptocurrencies in the banking sector).

However, the extent to which that development toward becoming a digitalized company will continue cannot be determined based on the industry sector alone. The different types of value creation and other factors, including specific regulations in the banking sector or dependencies in the manufacturing industry, do not necessarily influence the current state of digitalization but do influence the type of technologies used and their method of implementation. Saam et al. [22] concluded that, due to a lack of digitalization strategies, the majority of German SMEs are not yet engaging the process of digital transformation. However, according to our results, that conclusion does not or no longer applies. All 14 SMEs exhibited changes and developments in aspects of digitalization and thus seemed to have already initiated the process of digital transformation, albeit at different stages.

B. Differences between the companies

Differences also surfaced between the individual SMEs in the sectors examined. For one, a direct correlation between the number of employees and the perceived sources of potential and challenges, as well as the forms of the implementation of digital technologies, cannot be derived from the interview results. Current and future opportunities are also limited by financial, cultural, and business model-specific factors that do not necessarily correlate with company size. Pioneers in using new technologies and automation, including the automotive electronics manufacturer N and the direct banks E and K, even have comparatively small staffs, possibly due to a higher degree of digitalization and automation. Compared with the other companies interviewed, those digital pioneers have clearly already recognized the added value of digitalization for their business models, as was the case with the automotive electronics manufacturer N and the B2B manufacturer of fully automatic coffee machines C. Otherwise, their business models have always been based on digital technologies, as in the case of the direct banks (E and K). Other medium-sized manufacturing companies—for example, the metal industry company F and the furniture manufacturer G—lack the necessary production size or quantity of identical parts relative to the pioneers, despite the need for such parts in order to standardize and automate their production and thereby create a basis for Industry 4.0 concepts. In addition to a lack of starting points for automation, financial aspects could also discourage efforts at strategic automation in medium-sized companies. Expert N, for instance, reported that a fully automated production line would require investments ranging in the tens of millions.

C. Effects of technologies used

None of the interviewees indicated a clear distinction between data-driven processes and data-driven decisions, most likely because similar data foundations often form the starting point for operational and strategic decisions. At the same time, many data analytics are examples of automation and the provision of a digital IT infrastructure and enable virtual collaboration among employees and with partners and customers. Various cases revealed that digitalization in companies usually empowers but does not automatically support or stimulate change or facilitate work, regardless of whether it involves technological infrastructure or applications. For example, whereas current software systems offer many opportunities to support work processes, systems that have been implemented in ill-conceived ways can ultimately create more work and undermine efficiency (Experts H and M).

According to Expert G, more extensive databases enable deeper analyses but do not automatically simplify the understanding of one's company due to the increased complexity entailed. According to [23] and Expert N, although increased networking and automation as part of Industry 4.0 enable flatter hierarchies, and although all of the companies interviewed are using IT that simplifies exchange between hierarchical levels, in no case have those trends automatically yielded a flatter organizational structure or cross-hierarchical collaboration. On the contrary, changes in management and corporate culture are seen as triggers for those developments (Experts A, B, and J). That view of technology as purely an enabler redoubles the emphasis on positioning people as decision-makers, installers, and users of the technology.

V. Discussion

In what follows, four theses consolidated from the results of the literature review are discussed and differentiated against the background of the findings from the expert interviews.

Thesis 1: *The virtualization of processes and simplified access to software and data via cloud computing and private devices give employees greater flexibility regarding the time, place, and design of their work. However, a lack of trust and of control options continues to decelerate that development. As a result, the boundaries between work and private life are becoming increasingly fluid.*

All experts described increased flexibility in their day-to-day work and throughout their companies as being a result of digitalization efforts. To achieve those assets, all companies enable data access beyond their company sites via not only cloud solutions but also VPNs or remote desktop connections. Thus, cloud solutions have been primarily used in the SMEs interviewed (i.e., 12 of the 14 SMEs) to boost the flexibility of work. The two remaining companies reported currently debating whether they would implement cloud solutions.

At the same time, from a technological viewpoint, all companies afford the option of working from home, although such potential is not being exploited in every company interviewed. On the one hand, that tendency is due to activities in production and/or customer consulting, which are tied to fixed workplaces and times due to their work content. Thus, even for work activities that could be performed outside the company sites, the option of using a home office is sometimes waived by employees. One reason could be that employees of SMEs, as in the example of direct bank E, often have short travel distances and enjoy working together in person (Expert E). However, the interview results, as the academic literature similarly shows, also indicate a lack of trust on the part of managers as a reason why employees are less likely to work from home than they would like (Experts K and M). For that same reason, the spread of trust-based working hours also seems to have been inhibited; thus far, employees primarily at higher levels of the organizations' hierarchy have pursued the possibilities of remote work and trust-based working hours. At lower levels of the hierarchy, by contrast, the unverifiable nature of remotely performed services is presumably the most important cause for distrust on the part of managers, for work performance in most companies continues to be determined in hours, not according to measurable results. However, as per some interviewees (Experts I and N), the COVID-19 pandemic has spurred rethinking about how work can be performed, as many companies became forced to switch to work-from-home models and thus gained experience with digitalization.

All of the interviewed experts stated that the boundaries between their personal work and private lives were becoming increasingly blurred. However, that tendency is largely justified by their professional development and the responsibility borne within the scope of their current jobs, not by new technological opportunities available to them. Another reason given was that companies respect their employees' personal lives and do not require them to be continuously available or to work beyond business hours (Expert D). On the whole, the interviews suggested that employees themselves can somewhat influence the separation of work and private life, depending on how strongly the companies support such separation and how much the employees prefer to work beyond the work hours required (Expert K).

Thesis 2: *The further development of IT and increasing spread of social networks simplify communication within and between companies. However, the resulting decrease in personal interaction makes that communication less clear and team-internal coordination more complex.*

All of the companies interviewed offer a wide range of communication channels, such that contact increasingly occurs not only in person but also via email, telephone, video conference, intranet, and/or collaborative documents. However, in some companies, the potential of digital communication channels remains limited by a lack of sufficient Internet access (Experts H and L). At the same time, despite their theoretical potential as one such channel, social networks were not reported to be relevant in the experts' professional contexts, largely owing to the uncertain safety of exchanging business data via those channels and the preference for secure, direct means of communication (Experts M and H).

Overall, however, the share of virtual communication has increased in both industry sectors investigated, both for coordination within teams and throughout companies and to reduce the effort and cost of face-to-face meetings and central events. However, the greater flexibility possible in selecting personnel reported in the literature [24] was not characterized by interviewees as an advantage of working in virtual teams, partly due to the low degree of internationalization and the focus on local employees. Digital channels are also becoming increasingly important in customer–company contact, especially after initial personal contact, for they save time and costs and can create new offerings—for example, video consulting with banks and various services in the manufacturing industry. That development will continue but not completely replace personal contact, the experts unanimously agreed, given the importance of informal communication and the building of trust (Experts A, G, and M).

The experts' statements also confirm a real or feared decrease in uniqueness in the results of work due to intense virtual communication. Even so, the experts did not describe more complex internal team coordination as being a challenge. Expert A suggested that different personality types might also influence the preferred way of working together, an assumption supported by past research [25] showing that individuals most suited to virtual collaboration are ones who are open to new environments and who prefer short, targeted discussions and rapid decision-making. The results of that study also suggest that extroverts prefer to collaborate face-to-face but prefer virtual teams to working independently, whereas introverts can adapt more quickly to work on virtual teams because they have to expend less energy than in face-to-face interactions. Those deductions emphasize the importance of personality and cognitive style as factors influencing the success of working in virtual teams and at home offices.

Thesis 3: *Via digitalization, internal and external ideas (e.g., from customers) to the company can be more easily created, tested, integrated into innovations, and scaled in digital form. Nevertheless, bringing innovations to market remains complex.*

According to the interviewees, ideas for innovative projects have come from all departments and hierarchical levels in their SMEs, some of which have been technologically supported by supplying idea management platforms that simplify the collection and evaluation of ideas (Experts A, M, and N). In both industries, increased collaboration between departments and permeability between hierarchical levels can be observed, which has consequently increased the importance of ideas from all employees compared with ideas from individual decision makers and individual departments (e.g., R & D). Companies such as Volksbank A have also enabled and explicitly encouraged their employees to share their ideas (Expert A).

To date, the theoretical potential of digital innovations has had only a minor impact on manufacturing in the B2B sector, where the focus continues to be the further development of haptic core products instead. To test those products, prototypes are already being created in some of the SMEs using 3D printers; however, they have neither been fixed components of innovation processes nor benefited series production. Only a few companies already offer supplementary digital services along with their core products, possibly due to the technical complexity of such development (Experts F and G) and/or a lack of demand in the B2B market. However, similar to 3D printing, complementary services are seen as having tremendous potential in the future.

Thesis 3 applies more strongly to the banking sector. In recent years, the demand for digital services and consulting offerings has increased, as has customers' use of online branches and banking apps, such that the potential for digital innovations has also increased beyond purely in-house solutions. The banking interviewees reported that their organizations were digitizing their core products and services, whereas doing so is more difficult, if not impossible, in the manufacturing industry. In that context, the importance of IT providers also becomes apparent. Excluding the direct banks, the smaller banks in the sample were described as using central IT service providers, which are thus responsible not only for managing the central IT infrastructure but also for creating central, digital product and service innovations in collaboration with the banks (Experts A, B, and D).

By contrast, the model for success of fintech companies is to create quickly scalable, innovative solutions and platforms in a short period, which requires very little infrastructure and precludes having to meet as many regulatory requirements as companies with a banking license would have to meet (Expert H). However, in many cases, fintech companies also need partners, either to test solutions or to integrate them with services from other providers. Collaborating with fintech companies also plays an increasingly important role for banks such as Volksbank A and direct bank E, as mentioned by their respective experts.

***Thesis 4:** The impact of digital technologies on the labor market is primarily limited to the loss of repetitive, clearly defined activities, not entire professions. Moreover, new activities and professions are created through their use. Advances in artificial intelligence (AI), however, could increase possibilities for automating activities that are more demanding.*

Although the greatest changes due to automation have occurred in simple production and service activities, none of the companies interviewed have had to significantly reduce the scope of the jobs affected. Instead, for example, the B2B manufacturer of fully automatic coffee machines C made the strategic decision to not further automate certain work processes (Expert C). Along with the resources available to SMEs for technical automation, other factors—the availability of labor, the business model, and measures taken to expand and enhance employees' skills—can also reduce the extent of digitalization's effects on employment. In line with past deductions [16], the interviewed experts agreed that new tasks and added value for employees have to be found or developed as automation intensifies and that willingness and ability to change are important competencies to that end.

The changes described in the everyday lives of the experts also suggest that automation in SMEs has already reached impressive heights, at least in certain business areas, and is increasingly becoming the standard. However, that no expert reported having more time due to using digital technologies also implies that the changes triggered by automation do seem not permanent. Instead, outsourcing one's activities to software or machines becomes the standard after a certain time, and the time saved is continuously replenished with new tasks. Those factors could explain why some experts struggled to identify and describe automated processes in their respective environments. Regarding the manufacturing sector, the interviews revealed trends that reflect past findings ([26], [27]), namely that collecting data and using digital technologies have so far been aimed at controlling and optimizing plants, not automating or autonomizing them. However, as interviewees at the automotive electronics manufacturer N and plastics industry company J revealed and in contrast to published findings ([26]), investments are being made in not only lower-cost technologies such as cloud computing but also in innovative, automated production lines (Experts J and N).

The second claim of the thesis, concerning newly created jobs, can be justified by two effects, as the results of the interviews suggest. On the one hand, using digital technologies can secure the competitiveness of companies and thus company growth and jobs (Experts H and N). On the other, two-thirds of the experts (Experts A–E, I, K, L, and N) occupy positions directly created by digital technologies in recent years and that involve dealing with them. For that reason, the SME representatives interviewed seemed largely aware of digitalization's operational and strategic importance, and their organizations seem to be increasingly embedding it in their operations. On top of that, the jobs created exemplify professions with job profiles described by the experts as being less susceptible to automation and thus as the most important fields of human activity. Those activities involve monitoring and shaping digitalization, making strategic decisions and assuming responsibility, and engaging in activities characterized by a high degree of trust and communication.

Regarding the third claim of the thesis, the expert interviews provided little evidence that learning, artificially intelligent systems already play a role in the automation of demanding activities or even entire professions at SMEs. In the case of direct bank E, the system used to check credit applications is largely a rule-based activity, and the AI system for fraud prevention at direct bank K analyzes transactions executed on the basis of rules as well, albeit without recommending or initiating action independently (Experts E and K). The low prevalence of advanced AI systems could stem from the lack of areas of application, technological competence, and sufficient databases in SMEs for deep learning algorithms (Expert D). In the future, the first systems for more complex application areas are planned with chatbots for automatic interaction with customers (Expert I) and visual control of surfaces in production processes (Expert J). Even technically skilled interviewees such as Experts D and K attested to the limits of AI systems in human activities that are difficult to automate and need an extent of creativity that often cannot be provided or covered by such systems.

VI. Recommendations for Action

In what follows, to help SME managers to tap into the identified potential of digital technologies and to overcome the challenges associated with that endeavor, we make specific, cross-industry recommendations for action in four domains of activity.

First, contact restrictions during the COVID-19 pandemic have again shown that companies accommodating time- and location-independent work can respond more flexibly to changes. The pandemic has had positive effects on collaboration and future ways of working as well, in addition to having catalyzed virtualization and the accelerated transformation of organizations via digitalization [28]. According to Expert N, SMEs should primarily allow employees to work from home if they are well suited to that way of working and if it suits their job profile. New forms of measuring performance unrelated to hours worked and the use of improved digital reporting and monitoring tools could boost trust in the effectiveness of remote employees. As in the customer–company relationship, trust in teams should nevertheless initially be built through personal contact before communicating predominantly via virtual channels. Plus, at that point, video instead of telephone conferences are recommended for sustained virtual contact.

Second, to manage the complexity and speed of technological development, SMEs should develop a data and digitalization strategy with specific steps. The results of the expert interviews indicate that complex technologies such as distributed ledger technologies and AI applications will become increasingly important for SMEs in the future. For that reason, SMEs should begin examining possible applications and strategic roadmaps in order to avoid missing the starting gun for a successful digital transformation.

Third, wherever possible, operational and strategic decisions should be supported by data analyses in order to increase the quality of decision-making. Because an insufficient basis of data was often a challenge for the SMEs, rapid, digital testing methods should be used to expand that basis and/or to verify decisions. Especially when no decision-relevant data are available and speed is a pivotal factor, decisions should be made intuitively and in consultation with knowledgeable employees.

Fourth, in the SMEs interviewed, losses of employment due to automation had occurred in only a few cases, which can be attributed to active measures in addition to company growth. Companies that want to retain employees whose jobs have been adversely affected by digitalization should, similar to cooperative bank D, facilitate trial work and transfers to other jobs and departments (Expert D) and/or invest in the further training and retraining of those employees at an early stage. Another option is to prepare a skills matrix that enables the targeted deployment of employees in other activities according to a job rotation model (Expert C) or the expansion of the previous activity to include suitable, more automation-resistant tasks according to a job enrichment model.

VII. Conclusion and Further Research

The aim of our work was to identify the influence of digital technologies and trends on how companies work and what specific sources of potential and challenges SMEs have faced in the digital transformation. Therefore, as a primary part of our study, 14 semistructured expert interviews were conducted with executives at SMEs in the banking and manufacturing sectors, which we later systematically analyzed.

Altogether, the results suggest that digitalization has already triggered an array of changes in how companies work and will continue to do so in the future. The individual extent of the changes depends on, among other things, the industry sector, the business model, and degree of digitalization, as well as the job of the employee concerned. Therefore, our observations and the observations of the experts should always be interpreted in their respective contexts. Considering those factors, the findings from the scientific literature can largely be transferred to practice in SMEs. In the companies interviewed in both sectors, digitalization has prompted new forms of decision-making, increased flexibility in the choice of where and when to work, a change in how ideas and innovations are handled, new digital forms of communication and collaboration among employees or with customers, and the increased automation of simple, repetitive activities. The work methods of the experts interviewed, whose activities as managers are characterized by significant complexity and responsibility, have also changed as a result of digitalization. On average, their professional tasks have become more demanding and varied in recent years, and digital technologies have enabled the more efficient, more flexible, and sometimes more creative processing of tasks. As a result, instead of the experts' having more free time available, the activities performed have come to be perceived as denser. Added to that, the experts believe that the boundary between work and private life is becoming increasingly blurred, although that finding could be due to the experts' professional backgrounds.

Sources of potential and challenges amid digitalization pinpointed in the scientific literature were also largely confirmed by the experts' specific experiences and statements about future projects in their companies. The interviews revealed that digitalization, as a whole and in each of its technological trends, is associated with both opportunities and risks for companies. The experts' assessments of their respective companies differed only slightly depending on the industry sector. In both sectors, however, important sources of potential mentioned were greater flexibility with location- and time-independent work, easier communication via digital channels, better operational and strategic decisions, and more individualized customer-oriented offerings and approaches thanks to data analysis, as well as greater compliance and efficiency due to the automation of processes. Theoretical sources of potential for IT consumerization, by contrast, were not widely perceived by the SME personnel interviewed; however, increased sources of potential and challenges for Industry 4.0 were anticipated in the future due to the developing status of the companies. Challenges that were perceived by the experts included concerns with data protection, the lack of network expansion in rural regions, a decline in personal interaction, problems with data collection and analysis, high investment costs for digital technologies, and a great demand for further training to improve specific digital know-how within the companies.

Major differences in the type of technology used and the way in which it is implemented can be attributed both to external factors (e.g., regulations in the banking sector) and internal factors (e.g., forms of value creation and the business model used). Therefore, challenges with implementation for SMEs include access to sufficiently large amounts of data and corresponding specialists in the banking industry able to use predictive analytics and higher-performance deep learning algorithms, as well as the automation of production facilities if their volumes are too low or their product varieties too large.

In contribution to the current state of research on the effects of digitalization, this paper has provided, via expert interviews in two industry sectors with great potential for change, in-depth practical insights into specific areas for the application of digital technologies and into their effects. The differences between the selected industry sectors, as well as between the SME personnel interviewed, represent a broad scope of investigation. The results of the interviews specifically supplement literature on the state of research in the field with experience reports and assessments from an underrepresented viewpoint in the scientific literature: the viewpoint of SMEs.

It was not our aim to generate results that could be regarded as representative of the entire market. The industry sectors sampled were selected for their high potential of automation but are not representative of all SMEs in Germany. In some cases, due to differences between the sectors or the diversity within the manufacturing industry, a comparison within the same sector is possible only under certain conditions. For that reason, the results of our study cannot be readily applied to other industries. Further research could repurpose those limitations as an impetus to examine the transferability of the results, first to related industries (e.g., the insurance industry or manufacturing in the B2C sector) by using a similar research approach and, in turn, to other industries such as retail, agriculture, or health care. A larger, more diverse empirical approach would also allow determining the influence of factors such as company size, business model, or financial resources available. It is also expected that, due to the selection criteria chosen and self-selection effects, the experts in our panel were particularly familiar with digital technologies, and their assessments are therefore subject to a certain bias. The assessments of employees without managerial responsibility or representatives of other company departments (e.g., human resources, operations, finance, and logistics) in the manufacturing sector could be determined only indirectly in our study. Further research could thus involve supplementary interviews and focus specifically on the differences between the different corporate divisions and hierarchical levels.

The insights gained into digital technologies also offer starting points for further research. Future studies could, for example, examine the impact of technologies currently still in development, including blockchain and cryptocurrencies in the banking industry and autonomous driving in the manufacturing industry. Initial findings from business-focused surveys conducted around the world indicate that up to 70% of SMEs have intensified their use of digital technologies as a result of the COVID-19 pandemic [29]. Therefore, the pandemic's influence on the acceptance of remote work or the spread of virtual teams also presents an interesting area of research. In all, the specific sources of potential for using digital technologies to solve the current and future challenges identified highlight the relevance of further research on digital transformation.

Appendices

Appendix A: Interview guide

Question Block A: General questions

1. Do you see digital transformation as an opportunity or a risk for your company?
2. Which business unit of your company is most affected by digital transformation?
3. How do you assess the relevance of market changes and accommodate them in your company's or your own business activities?
4. Has digitalization led to changes in your company's hierarchy?
5. Which effects do IT and digital technologies have on your company's innovation processes?

Question Block B: Flexibility via digitalization

6. Has IT made you and your company more flexible overall?
7. Do you know about and use cloud services?
8. Private mobile devices and networks are increasingly used for professional purposes. What do you think about that development?
9. Do employees in your company work together virtually, and, if so, are there cases in which virtual interaction is the only form of interaction?

Question Block C: Data analysis

10. What significance do data and data analytics have in your everyday work life?
11. Are your company's IT infrastructure and organizational structure designed for increasing data volumes and data analysis?
12. Do you use any forms of technology to improve your strategic decision-making?

13. Artificial intelligence is developing rapidly and considered to have great potential. Have you already explored areas where AI could be applied in your company?

Question Block D: Automation

14. Has your company gained any experience with process automation?
15. Would you say that support with technological processes allows you to
    a) have more time?
    b) perform more creative tasks?
    c) have a more varied job?
    d) work on more demanding tasks?
16. Are you worried about losing your job as a result of automation?
17. What meaning does "Industry 4.0" have for you and your business sector?
18. What role do you see for employees in the future of Industry 4.0?
19. Which other digital technologies will alter how people work in the future?

Appendix B: Experts and companies interviewed

TABLE I.
OVERVIEW OF THE INTERVIEWEES

Interviewees		Companies		
Label	**Job title**	**Type and label**	**Number of employees**	**Industry**
Expert C	Chief Digital Officer	B2B manufacturer of fully automatic coffee machines C	400–450	Manufacturing
Expert F	General Manager	Metal industry company F	350–400	
Expert G	General Manager	Furniture manufacturer G	150–200	
Expert J	General Manager	Plastics company J	150–200	
Expert L	Innovation Project Manager	Manufacturer of measurement technology L	50–100	
Expert M	Manager Marketing and Communication	Control solutions manufacturer M	350–400	
Expert N	Corporate Development Manager	Automotive electronics manufacturer N	250–300	
Expert A	Head of Strategic Further Development	Volksbank A	150–200	Banking
Expert B	Head of the Digital Business Model Department	Sparkasse B	300–350	
Expert D	Senior Expert Innovation Technology	Cooperative bank D	650–700	
Expert E	Chief Digital Officer	Direct bank E	150–200	
Expert H	Head of Strategic Partnerships	Factoring bank H	250–300	
Expert I	Chief Digital Officer	Sparkasse I	500–550	
Expert K	Ex-Head of APIs and Open Banking Platforms	Direct bank K	300–350	

REFERENCES

[1] K. G. Gökçe and O. Dogerlioglu, " 'Bring your own device' policies: Perspectives of both employees and organizations," *Knowl. Manag. E-Learn. Int. J.*, vol. 11, no. 2, pp. 233–246, 2019, doi: 10.34105/j.kmel.2019.11.012.

[2] C. Leyh, T. Schäffer, K. Bley, and S. Forstenhäusler, "Assessing the IT and Software Landscapes of Industry 4.0-Enterprises: The Maturity Model SIMMI 4.0," in *Information Technology for Management: New Ideas and Real Solutions*, vol. 277, E. Ziemba, Ed. Heidelberg, New York: Springer, 2017, pp. 103–119. doi: 10.1007/978-3-319-53076-5_6.

[3] M. Pagani, "Digital Business Strategy and Value Creation: Framing the Dynamic Cycle of Control Points," *MIS Q.*, vol. 37, no. 2, pp. 617–632, 2013, doi: 10.25300/MISQ/2013/37.2.13.

[4] C. Leyh, S. Martin, and T. Schäffer, "Analyzing Industry 4.0 Models with Focus on Lean Production Aspects," in *Information Technology for Management. Ongoing Research and Development*, vol. 311, E. Ziemba, Ed. Cham: Springer, 2018, pp. 114–130. doi: 10.1007/978-3-319-77721-4_7.

[5] S. Mathrani, A. Mathrani, and D. Viehland, "Using enterprise systems to realize digital business strategies," *J. Enterp. Inf. Manag.*, vol. 26, no. 4, pp. 363–386, 2013, doi: 10.1108/JEIM-01-2012-0003.

[6] K. Bley, C. Leyh, and T. Schäffer, "Digitization of German Enterprises in the Production Sector - Do they know how 'digitized' they are?," in *Proceedings of the 22nd Americas Conference on Information Systems (AMCIS 2016)*, 2016.

[7] L. LaBerge, C. O'Toole, J. Schneider, and K. Smaje, "How COVID-19 has pushed companies over the technology tipping point—and transformed business forever," McKinsey & Company, 2020.

[8] R. F. Zammuto, T. L. Griffith, A. Majchrzak, D. J. Dougherty, and S. Faraj, "Information Technology and the Changing Fabric of Organization," *Organ. Sci.*, vol. 18, no. 5, pp. 749–762, 2007, doi: 10.1287/orsc.1070.0307.

[9] P. Daugherty and M. Carrel-Billiard, "The Post-Digital Era is Upon Us: Are you ready for what's next?," Accenture, 2019.

[10] J. Bughin *et al.*, "Tech for Good: Smoothing disruption, improving well-being," McKinsey Global Institute, 2019.

[11] C. B. Frey and M. A. Osborne, "The future of employment: How susceptible are jobs to computerisation?," Oxford Martin Programme on Technology and Employment, 2013.

[12] E. Henriette, M. Feki, and I. Boughzala, "The Shape of Digital Transformation: A Systematic Literature Review," in *Proceedings of the 9th Mediterranean Conference on Information Systems (MCIS 2015)*, 2015.

[13] P. Parviainen, M. Tihinen, J. Kääriäinen, and S. Teppola, "Tackling the digitalization challenge: how to benefit from digitalization in practice," *Int. J. Inf. Syst. Proj. Manag.*, vol. 5, no. 1, pp. 63–77, 2017, doi: 10.12821/ijispm050104.

[14] F. Nwaiwu, "Review and Comparison of Conceptual Frameworks on Digital Business Transformation," *J. Compet.*, vol. 10, no. 3, pp. 86–100, 2018, doi: 10.7441/joc.2018.03.06.

[15] I. Mergel, N. Edelmann, and N. Haug, "Defining digital transformation: Results from expert interviews," *Gov. Inf. Q.*, vol. 36, no. 4, 2019, doi: 10.1016/j.giq.2019.06.002.

[16] H. Bonin, T. Gregory, and U. Zierahn, "Übertragung der Studie von Frey/Osborne (2013) auf Deutschland," *ZEW - Zentrum für Europäische Wirtschaftsforschung GmbH*, Mannheim, Kurzexpertise No. 57, 2015.

[17] D. H. Autor, "Why Are There Still So Many Jobs? The History and Future of Workplace Automation," *J. Econ. Perspect.*, vol. 29, no. 3, pp. 3–30, 2015, doi: 10.1257/jep.29.3.3.

[18] K. Yilmaz, "Comparison of Quantitative and Qualitative Research Traditions: epistemological, theoretical, and methodological differences," *Eur. J. Educ.*, vol. 48, no. 2, pp. 311–325, 2013, doi: 10.1111/ejed.12014.

[19] J. Recker, *Scientific Research in Information Systems*. Berlin, Heidelberg: Springer, 2013. doi: 10.1007/978-3-642-30048-6.

[20] A. Bogner and W. Menz, "Das theoriegenerierende Experteninterview," in *Das Experteninterview*, A. Bogner, B. Littig, and W. Menz, Eds. Wiesbaden: VS Verlag für Sozialwissenschaften, 2002, pp. 33–70. doi: 10.1007/978-3-322-93270-9_2.

[21] R. Buber and H. H. Holzmüller, Eds., *Qualitative Marktforschung*. Wiesbaden: Gabler, 2007. doi: 10.1007/978-3-8349-9258-1.

[22] M. Saam, S. Viete, and S. Schiel, "Digitalisierung im Mittelstand: Status Quo, aktuelle Entwicklungen und Herausforderungen," *ZEW - Zentrum für Europäische Wirtschaftsforschung GmbH*, Mannheim, Forschungsprojekt im Auftrag der KfW Bankengruppe, 2016.

[23] P. Trompisch, "Industrie 4.0 und die Zukunft der Arbeit," *E Elektrotechnik Informationstechnik*, vol. 134, no. 7, pp. 370–373, 2017, doi: 10.1007/s00502-017-0531-1.

[24] T. L. Griffith, J. E. Sawyer, and M. A. Neale, "Virtualness and Knowledge in Teams: Managing the Love Triangle of Organizations, Individuals, and Information Technology," *MIS Q.*, vol. 27, no. 2, pp. 265–287, 2003, doi: 10.2307/30036531.

[25] A. Luse, J. C. McElroy, A. M. Townsend, and S. DeMarie, "Personality and cognitive style as predictors of preference for working in virtual teams," *Comput. Hum. Behav.*, vol. 29, no. 4, pp. 1825–1832, 2013, doi: 10.1016/j.chb.2013.02.007.

[26] A. Moeuf, R. Pellerin, S. Lamouri, S. Tamayo-Giraldo, and R. Barbaray, "The industrial management of SMEs in the era of Industry 4.0," *Int. J. Prod. Res.*, vol. 56, no. 3, pp. 1118–1136, 2018, doi: 10.1080/00207543.2017.1372647.

[27] W. Bauer, S. Schlund, T. Hornung, and S. Schuler, "Digitalization of industrial value chains – a review and evaluation of existing use cases of Industry 4.0 in Germany," *LogForum*, vol. 14, no. 3, pp. 331–340, 2018, doi: 10.17270/J.LOG.2018.288.

[28] O. Pakos, J. Walter, M. Rücker, and K.-I. Voigt, "The Leap into the New Normal in Creative Work: A Qualitative Study of the Impact of COVID-19 on Work Practices in Industrial Companies," *Eur. J. Bus. Manag.*, vol. 13, no. 10, 2021, doi: 10.7176/EJBM/13-10-01.

[29] C. Riom and A. Valero, "The business response to Covid-19: The CEP-CBI survey on technology adoption," Centre for Economic Performance, London School of Economics and Political Science, London, 2020.

DOI: 10.15439/2021F70
ISSN 2300-5963 ACSIS, Vol. 25

Risks of Concurrent Execution in E-Commerce Processes

Janis Bicevskis, Anastasija Nikiforova, Girts Karnitis
[[0000-0001-5298-9859, 0000-0002-0532-3488, 0000-0002-7563-6383]
Faculty of Computing University of Latvia
{Janis.Bicevskis, Anastasija.Nikiforova, Girts.Karnitis}@lu.lv

Ivo Oditis, Zane Bicevska
[0000-0003-2354-3780, 0000-0002-5252-7336]
DIVI Grupa Ltd
{Ivo.Oditis, Zane.Bicevska}@di.lv

***Abstract*—The development of ICT facilitates replacing the traditional buying and selling processes with e-commerce solutions. If several customers are served concurrently, e.g. at the same time, the processes can interference each other causing risks for both the buyer and the seller. The paper offers a method to identify purchase/sale risks in simultaneous multi-customer service processes. First, an exact model of buying-selling processes is created and the conditions for the correct process execution are formulated. Then an analysis of all the possible scenarios, including the concurrently executed buying-selling scenarios, is performed using a symbolic execution of process descriptions. The obtained result allows both the buyer and the seller to identify the risks of an e-commerce solution.**

***Index Terms*—concurrent processes, risk analysis, e-commerce.**

I. Introduction

THE development of ICT has created preconditions for radical changes in buying and selling processes worldwide. The traditional buying/ selling process, when sellers and buyers meet each other in person and communicate directly, is replaced by a remote communication, the so-called e-commerce.

Traditionally the buyer chooses a suitable product, checks its quality, receives a product, and pays for it directly to the seller (in cash or using safe bank services) in a shop, and no additional tools are needed. The advantages of e-commerce lie in the global spread allowing entrepreneurs to develop remote marketing and sales on an unlimited geographical scale. E-commerce solutions are perceived as the future of commerce, as more and more customers want to buy products without leaving their places [1]. By switching to remote seller-buyer communication, the information exchange may become more complicated and, at the same time, riskier for both the seller and the buyer. The seller is not willing to risk by sending the product to the buyer before being sure that the purchase is paid for; the buyer does not want to risk paying for a product is not seen yet and whose quality he has not been able to be sure of. Additionally, the processes are complicated because of many different product delivery channels and product payment options as well as by many simultaneous customers.

The main task of this research is to analyze the risks of remote buying and selling processes when e-commerce solutions are in use and simultaneous service of several customers take place.

This paper contains an analysis of several e-commerce cases that are common in different industries: theater ticketing, online stores, hotel reservation systems. The algorithm, applied for the analysis of process correctness, has been developed through a theoretical research for correctness of concurrent process execution [2]. The proposed algorithm identifies the possibility of incorrect execution of remote purchase-sale processes, thus revealing the risks for both buyers and sellers.

This paper is structured as follows: the background (Section 2), risk analysis of selected e-commerce processes (Section 3), analysis of the proposed solution (Section 4), conclusions and the future work (Section 5).

II. Theoretical Background

II.A. State of the Art

A literature review reveals different internet shopping-related risk classifications established over the last decades. A total number of risks considered to have a significant impact on users' intention for online shopping vary from one study to another and ranges from two to eight [3].

Forsythe et al. [4] identified six types of perceived risks that may have a negative impact on the experience of buyers: financial, product performance, social, psycho-logical, physical, and time loss. Respondents found financial risk to be the most important and significant. Considering the age of this study and the development of e-commerce over the past years, most of the identified risks have already been processed and resolved. However, some of them remain valid, for instance, financial risks.

Formerly, financial risks were primarily associated with potential losses of money due to fraudulent misuse of credit card information. Nowadays, the paradigm on financial risks has changed [5]-[6]. The online credit card usage-related risks are thoroughly discussed in security-related studies,

The research leading to these results has received funding from the research project "Competence Centre of Information and Communication Technologies" of EU Structural funds, contract No. 1.2.1.1/18/A/003 signed between IT Competence Centre and Central Finance and Contracting Agency, Research No. 1.6 “Concurrence analysis in business process models ".

and practical solutions are invented in online shopping platforms, including the implementation of 128-bit RSA encryption, digital certificates, firewalls etc. [6]. Another financial side-related risk is less covered: the trust between the customer and the service or/ and shopping service provider [5]-[8]. Trust and reputation are considered as the concepts dominating in e-commerce most [9], now. Bezes [10] proposes a classification where the probability of bank or personal data being stolen is understood as a "transaction risk". The probability of losing money when buying from an online store is defined as a "financial risk". There are studies rejecting the significance of financial risks, e.g. [11]. Based on a survey that has been carried out between 245 country residents, the study identifies the "convenience risk" and the "non-delivery risk" to be very significant, as well as the "reliability of shipper" and the "settling disputes". However, it should be mentioned that, other classifications consider both abovementioned risks as financial risks.

Another risk, namely "performance risk" is associated with the potential failure of a product or website to meet expected performance requirements, i.e., the uncertainty regarding the after-sales service [8], [12]. According to [12], "risk perceptions" and the "online shopping intention" have a significant impact on online shopping.

[3] and [7] revealed that risks such as privacy, source, performance, payment and delivery risks are predominant dimensions in Internet shopping. The authors of [7] have carried out an in-depth analysis of risk-relievers. Although only one rather limited example of shopping has been analysed with a sample size of 471 respondents, the authors' research suggests that 18 risk-relievers make sense for shopper. The main risk-relievers are (1) payment security, (2) money-back guarantee, as well as possibility (3) of exchanging the item, (4) of viewing the item, (5) of seeing item in a store, (6) price and (7) website reputation. [10] claims that guarantees provided by online sellers and insurance against any kind of adverse situations were assessed as the most important factor (88.7% of respondents) to drive online shopping.

In this study we provide a method that allows the identification of risks arising from the concurrent execution of processes without the e-commerce risks mentioned above. This topic is especially relevant due to COVID-19 pandemic as online shopping has become a daily phenomenon for most of the population, i.e. online stores have been launched in countries and cities where they did not exist before.

II.B. Analysis Basics

This study considers an algorithm for detection of incorrect concurrent execution of business processes that use a transaction mechanism. First, let's clarify the basic concepts, which detailed description is provided in [13].

The process will be defined as a set of actions described in a modeling language. Two levels of process descriptions are possible: (1) the model is informal without well-defined semantics of operations; (2) the process is described with program code in a programming language where the semantics of the actions are unambiguously defined. In the real world, many processes run concurrently, i.e., multiple instances of processes are executed simultaneously with different inputs, and shared information resources can be used. If several concurrently executed processes perform operations on the same data, then the data may be changed by another process during the breakpoint of one process, where a breakpoint is a process activity at which a process can be stopped and later restored from the state it was in before the break [13]. This can lead to incorrect system operation, which cannot happen if the processes are executed serially. Algorithms of business process analysis to determine the possibility of incorrect result of concurrent execution are the result of the theoretical research [2].

The method, proposed in [2], let us identify an incorrect result of concurrent execution of several processes. In the case of databases, we consider the concurrently executable processes Pj, Pk… Pm and define the correctness of their execution according to the DBMS ACID correctness [13]: the result is correct if one process is executed without simultaneous execution of other processes and all transactions within the process are executed serially. If several processes are executed serially, the result is also correct, though there may be several different but correct results, depending on the sequence of process execution. The exact criterion to achieve a correct result for any process and any input data is to execute processes serially.

II.C. Risk Analysis Algorithm

There are six main steps of the universal algorithm.

Step I: create a description of the business process. In order to analyze a business process, a model of a business process should be created. If the business process is described informally, the model can be designed as a graphical diagram where the vertices of the graph represent the activities of the business process and the arcs the sequence of activities. The model's author must be able to assess the feasibility of scenarios and the outcome of scenarios. If the model consists of a program code, the execution of statements and the sequence of activities are strictly defined. In this case, the code analysis can be performed automatically by a tool. The program code is executed symbolically: the tool compiles the conditions for the execution of the scenario and calculates the result of the execution of the scenario.

Step II: define business process transactions. Business process activities or a set of activities are defined as transactions in cases when their execution is delegated to another system (for example, to a DBMS) or their execution requires timeframe during which access to common resources may not be blocked for other processes. For example, the process of selling theater tickets can be divided into three transactions: (1) read from the database the seats sold, (2) let the client choose a free seat in the hall, (3) let the client pay for the selected tickets. The ticket selection process should not be blocked for other remote customers, as the selecting may take some time.

Step III: define the incorrect business process execution. This step identifies situations that are not acceptable from a business perspective. If the process execution scenario leads to a situation that does not meet the business requirements, then the definition of the business process needs to be revised to avoid incorrect execution results.

Step IV: construct a feasible scenario tree. The model's author selects different process execution scenarios and evaluates their feasibility; the author makes sure that there is input data that will make the selected scenario executable. The result of the analysis is represented in a tree, where each branch of the tree represents one feasible scenario and the tree contains all possible different scenarios. Depending on the business process this can be a difficult goal to achieve. If the model is defined by a program code, the "white box" analysis is used by symbolic execution of the program code, which enables to compile the conditions for the execution of a pre-defined scenario. When solving the conditions, the solution obtained is a test case that should be executed to cover pre-defined paths.

Step V: calculate scenario execution results. The model's author evaluates the expected result of the scenario execution from the business point of view using a symbolic execution.

Step VI: identify scenarios that lead to incorrect business process execution. According to [13], two sets of process execution scenarios are analyzed – a set of concurrent execution scenarios (C) and a set of serial execution scenarios (S). If at least one scenario S_j from S can be found for the scenario C_i from C such that the set of conditions and results of fulfillment of C_i coincides with the set of conditions and results of fulfillment of S_j, then the concurrent execution C_i is correct, otherwise it is incorrect.

III. A Proposed Solution

In this section, the algorithm that identifies risks in e-commerce processes will be applied to several e-commerce solutions: business processes for theater ticketing, online stores and hotel reservation systems. All these e-commerce cases consist of three steps: ordering a product/service, payment for the goods, and delivery of the goods. The steps vary depending on the industry and the implementation. We will identify risks for both the seller and the buyer using different purchase-sale scenarios. To simplify the analysis, the processes will first be considered for ticket sales, and then they will be modified for other sectors.

III.A. Internet Shop for Theatre Ticketing

In the past, theater tickets were sold at ticket offices, and customers were served in presence. The currently available ticketing systems offer e-commerce functionality: connect to the system remotely, select a ticket, pay for it and receive a copy of the ticket. The purpose of the following sections is to identify the potential risks posed by the concurrent service of several customers.

A.1. Defining of Incorrect Process Execution

The correctness of the ticketing system's performance will be assessed by the status assigned to the seats in a hall. The status *seatStatus* will be determined by the values of two parameters: *availability of a seat* – {*available, reserved, sold*} and *status of payment* = {*paid, not paid*}. The ticketing system works correctly if the attribute *seatStatus* for any seat in the hall has either <*available, not paid*> or <*sold, paid*> as values. Any other result shall be considered as incorrect. If there are seats with the status "*sold*" and "*not paid*" at the same time, then the ticket system works unacceptably.

Process execution scenarios will be analyzed below to determine if there are possible scenarios that could lead to incorrect results.

A.2. The First Phase of the Business Process: Select a Seat

The ticketing system consists of three sequential phases: *Select a seat*, *Pay for a ticket* and *Send a ticket*. The first phase *Select a seat* consists of three activities: *readSeats*, *selectSeats*, *reserveSeats*. The activity *readSeats* reads information from the database about the customer's chosen performance and, if the event is not sold out, shows it to the customer. The activity *selectSeats* allows the customer to mark the chosen seats in the halls plan. The activity *reserveSeats* changes the information on occupied seats in the database by assigning the value "reserved" to the seat.

The *selectSeats* operation can take a longer time and therefore, during its execution, the common resource may not be locked; the information about the seats should be available to other customers. All three operations - *readSeats*, *selectSeats*, *reserveSeats* - will be executed as separate transactions. Thus, the ticketing system can execute many transactions from different customers' business processes "simultaneously", ensuring the execution of successive transactions for each individual process.

Unfortunately, the simultaneous service of several customers can lead to incorrect execution of the process. Let us construct a concurrent execution scenario of two processes P_1 and P_2:

P1(readSeats,YES)=>P2(readSeats,YES)=>
P1(reserveSeats) => P2(reserveSeats)

This scenario is feasible but there is a risk of selling the same seat to two customers if customers from both processes P_1 and P_2 choose the same seat. This is unacceptable, and the simplified seat selection process is risky. The situation changes drastically when seat reservation is used: a control mechanism checks whether the seat is already booked by another process. A correct process model, in which the data on free seats in the hall are re-read before reserving a seat and in case the seat selected by P_1 is already reserved in another process P_2, the seat selection step is repeated. The business process is changed by adding additional controls before the actual reserving in *reserveSeats*.

A.3. The Second Phase: Pay for a Ticket

Banking systems offer many ways to pay for the tickets bought. In all cases, the step *Pay for a ticket* must be performed as a separate transaction, because the service of other customers may not be interrupted until the end of the ticket payment process.

Step *Paying for a ticket* poses risks to both the seller and the customer: the seller reserves tickets for a certain period, preventing them from buying other customers, and the customer, in turn, pays for the ticket, believing that he will receive the ticket on time.

It is even more difficult for the ticket system to get a secure ticket payment because it is done by an external (bank's) payment system. Different banks have different payment solutions, which makes it difficult for the ticket system to unify payment processes.

The phase *Pay for a ticket* contains three activities: *readAccount*, *checkValue* and *writeAccount*. The activity *readAccount* reads the customer's account balance from the bank's database, the activity *checkValue* checks whether the customer enough means to pay the ticket price.

If payment can be made, the activity *writeAccount* deducts the amount payable for tickets from the account balance and stores the new account balance in the bank system's database. Activities *readAccount* and *writeAccount* are executed as independent transactions. And it leads to risks that the payments may be executed incorrectly if run concurrently [2]. However, as concurrent payment execution from a common resource (from one bank account) for several customers is unlikely, there are grounds to assume that the *Pay for the ticket* transaction is executed as one indivisible transaction and it cannot affect the service of other customers.

A.4. The Third Phase: Send a Ticket

It is possible two situations: (1) the payment was not completed successfully - the ticketing system sends a message to the customer about the refusal to purchase tickets, the corresponding seats in the halls plan are marked as available by the activity *changeStatus*, and the seller may sell the ticket to another customer; (2) the payment was not confirmed timely - this situation may occur if the message has not been received from the payment system timely. Different solutions are possible: resend the invoice to the customer or cancel the purchase, mark the corresponding seats as available for resale (*changeStatus*). This solution runs the risk that the ticket is actually paid for, but the ticket is resold to another customer due to a delay in reporting.

Sending a ticket to the customer without receiving a feedback and relying on the stable operation of the Internet are debatable. This defect can be remedied by providing a confirmation of receipt of tickets sent by the customer.

III.B. Online Store

The operation of the online store is determined by four steps of the process - marketing, selection of goods, payment for goods and delivery of goods. We will not consider marketing issues in this paper, the other three steps are similar to the operation of ticketing systems. However, there are certain peculiarities of the industry in the processes of online stores, which are often related to the efficiency of delivery processes.

The activity *selectItem* differs from the choice of tickets significantly because the customer wants to choose the product personally, look at it and evaluate it in detail. The online store can only display similar samples from catalogues remotely. The customer will make the final assessment of the product only after receiving the product, when the product has already been paid for. If the payment for the goods is not prompt, the online stores can sell the goods to another customer who has paid for the goods faster. Such situations occur regularly in practice.

Even more risky is the customer's cooperation with the online store in cases when the online store orders goods only after placing a customer's order or payment. In other words, online stores without warehouses with stocks of goods are quite risky in terms of delivery. If the goods are not reserved upon receipt of the customer's order, the delivery of the goods to another customer who has made the payment earlier is not excluded, thus extending the delivery time.

The activity *payItem* does not differ significantly from the payment of tickets, however, additional risks are expected to be made if the amounts to be paid are significant and some purchases may lack money. The promptness of the payment has a significant impact on the process, as the delivery terms of the goods depend on it. Payment via Internet banking at the time of ordering (*Banklink*) is not only the safest, but also the most modern for shopping in online stores.

The risks of the activity *deliveryItem* are similar to those of ticketing processes – (1) the customer receives a product the quality of which has not been checked, (2) product delivery terms are determined by the customer's product payment efficiency and online store processes, (3) the online store can sell the product ordered by the customer to another customer who has paid for the product faster.

III.C. Hotel Reservation

The hotel reservation and payment processes are characterized by the fact that a hotel room is reserved for a specific customer for a specific timeframe. It is not possible to assign the same room to two customers, and it reduces the risks of the process. However, there is a risk that the room reserved for the customer will not be released in time, for instance, because the previous customer has extended his stay in a hotel. Such situations can be resolved only by a hotel staff.

An insignificant risk exists if the customer is trying to find the most advantageous hotel among others for a longer time. In the moment of booking, the special offer may no longer be valid because another customer has already booked it.

Payment for hotel services is usually made during the check-out process. The customer acknowledges the services received and pays for them. Unfortunately, the credit card may have not enough coverage to pay for the services received.

Summarizing the described process, it contains several risks caused by the concurrent process execution. These risks can be mitigated by identifying them in the information system and involving hotel staff.

IV. Analysis of The Proposed Solution

Summing up our experience, analysis of concurrently executable e-commerce processes is required in at least three cases:

1. for online store customers to identify the risks of purchasing: does the quality of the selected goods meet the customer's requirements? Will the goods be delivered on time? Is there a risk of non-delivering for pre-paid goods?
2. for online store owners to make sure the customer's solvency and reliability as well as to develop and improve the business processes;
3. for online store developers before the final implementation stage to detect vulnerabilities, defects, and errors in the business processes. This can save significant programming resources that could be wasted for implementing an incorrect business process (also in line with [13]).

V. Conclusion

The development of ICT has created preconditions for performing traditional buying and selling processes remotely using e-commerce solutions. The paper offers an algorithm for the analysis of e-commerce processes to identify the risk caused by their concurrent execution. The main steps of the algorithm:

- an accurate model of the e-commerce process is developed at such a level of accuracy that it is possible to determine the feasibility of any process scenarios and calculate the result of the scenario execution,
- process execution correctness conditions are formulated, and it allows each scenario to determine the compliance of its execution result with the process correctness conditions,
- the analysis of all different possible purchase-sale scenarios is performed using the symbolic execution of the process description, whereby concurrent execution of several customer processes is allowed. The obtained result of scenario execution allows identifying the process risks, which in turn allows both buyer and seller to choose an e-commerce solution acceptable to them.

The further development of the research can be devoted to additional e-commerce processes. The implementation risks should be analyzed, the processes improved, and as a result the use of modern e-commerce solutions would be expanded.

References

1. Ecommerce Guide, “Ecommerce Statistics”, online: https://ecommerceguide.com/ecommerce-statistics, last accessed: 21.05.2021
2. J. Bicevskis, G. Karnitis, “Testing of Execution of Concurrent Processes,” *In International Baltic Conference on Databases and Information Systems* (pp. 265-279). Springer, Cham, 2020, https://doi.org/10.1007/978-3-030-57672-1_20.
3. E. O'Callaghan and J. Murray, “The International Review of Retail, Distribution and Consumer Research,” *The International Review of Retail, Distribution and Consumer Research*, vol. 27, no. 5, p. 435–436, 2017 doi:10.1080/09593969.2017.139195
4. S. Forsythe and B. Shi. “Consumer patronage and risk perceptions in Internet shopping,” *Journal of Business research*, 56(11), 867-875, 2003.
5. R. M . Al-dweeri, Z. M. Obeidat, M. A. Al-dwiry, M. T. Alshurideh, A. M. Alhorani, “The impact of e-service quality and e-loyalty on online shopping: moderating effect of e-satisfaction and e-trust,” *International Journal of Marketing Studies*, 9(2), 92-103, 2017, http://doi.org/10.5539/ijms.v9n2p92.
6. R. Thakur, M. Srivastava, “A study on the impact of consumer risk perception and innovativeness on online shopping in India,” *International Journal of Retail & Distribution Management*, 2015, https://doi.org/10.1108/IJRDM-06-2013-0128.
7. P. Rita, T. Oliveira, A. Farisa, “The impact of e-service quality and customer satisfaction on customer behavior in online shopping,” *Heliyon*, 5(10), e02690, 2019, https://doi.org/10.1016/j.heliyon.2019.e02690Get rights and content.
8. N. Pappas, “Marketing strategies, perceived risks, and consumer trust in online buying behavior,” *Journal of retailing and consumer services*, 29, 92-103, 2016, https://doi.org/10.1016/j.jretconser.2015.11.007.
9. C. Sellami, M. Baron, M. Bechchi, A. Hadjali, S. Jean, D. Chabot, “Towards a Unified Framework for Computational Trust and Reputation Models for e-Commerce Applications,” *In Research Challenges in Information Science. RCIS 2021*, vol 415. Springer, Cham. https://doi.org/10.1007/978-3-030-75018-3_44, 2021
10. C. Bezes, “Comparing online and in-store risks in multichannel shopping,” *International Journal of Retail & Distribution Management*, 2016, https://doi.org/10.1108/IJRDM-02-2015-0019.
11. K. Wai, O. Dastane, Z. Johari, N. B. Ismail, “Perceived risk factors affecting consumers’ online shopping behavior,” *The Journal of Asian Finance, Economics and Business*, 6(4), 246-260, 2019, https://dx.doi.org/10.2139/ssrn.3498766.
12. U. Otika, E. Olise, O. B. Oby, “Risk Perceptions and Online Shopping Intention among Internet Users In Nigeria,” *Global Journal of Management And Business Research*, 2019.
13. A. Nikiforova, J. Bicevskis, G. Karnitis, “Towards a Concurrence Analysis in Business Processes,” *In 2020 Seventh International Conference on Social Networks Analysis, Management and Security (SNAMS)* (pp. 1-6). IEEE, 2020, DOI: 10.1109/SNAMS52053.2020.9336566

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 453–457

DOI: 10.15439/2021F44
ISSN 2300-5963 ACSIS, Vol. 25

Effect of Criteria Range on the Similarity of Results in the COMET Method

Andrii Shekhovtsov, Jakub Więckowski, Bartłomiej Kizielewicz and Wojciech Sałabun
Research Team on Intelligent Decision Support Systems,
Department of Artificial Intelligence Methods and Applied Mathematics,
Faculty of Computer Science and Information Technology
West Pomeranian University of Technology in Szczecin
ul. Żołnierska 49, 71-210 Szczecin, Poland
Email: {andrii-shekhovtsov, jakub-wieckowski, bartlomiej-kizielewicz, wojciech.salabun}@zut.edu.pl

***Abstract*—Defining input values in the decision-making process can be done with appropriate methods or based on expert knowledge. It is essential to ensure that the values are adequate for the problem to be solved in both cases. There may be situations where values are overestimated, and it should be checked whether this affects the final results.**

In this paper, the Characteristic Objects Method (COMET) was used to investigate the overestimation effect on the final rankings. The decision matrixes with a different number of alternatives and criteria were assessed The obtained results were compared using the WS similarity coefficient and Spearman's weighted correlation coefficient. The study showed that overestimation has a significant effect on the rankings. A larger number of criteria has a positive effect on the correlation strength of the compared rankings. In contrast, a large overestimation of characteristic values has a negative effect on the similarity of the results.

I. Introduction

In decision-making, expert knowledge is an important element influencing the results obtained [1]. It is important in specifying the importance of criteria and the weighting of each criterion in the process of evaluating alternatives [2], [3]. These decisions directly translate into the obtained preference values guaranteed by the selected multi-criteria methods [4], [5], [6].

For some Multi-Criteria Decision-Making (MCDM) methods to solve decision-making problems, the expert must define the algorithm's input parameters based on his experience and knowledge [7], [8]. Some methods allow the use of methods that determine weights for criteria in a defined problem [9], [10]. In other cases, the data determined for the method's operation must be specified solely based on expert knowledge [11], [12]. Multi-Criteria Decision-Making methods are eagerly used in solving problems where many factors contribute to the final assessment [13]. The development of new techniques attracts the attention of a growing audience, who use them to solve medical problems [14], [15], [16], [17], for resource planning [18], [19], [20], or the selection of sustainable means of transport [21], [22], [23].

One of the multi-criteria methods is the Characteristic Objects Method (COMET), which uses the rule-based approach when evaluating the quality of alternatives [24]. The expert's task using this method to solve the problem is to determine the characteristic values, which will be used to assess the preference of alternatives in subsequent steps [25], [26]. The advantage of this method is that it is resistant to the phenomenon of ranking reversal when the number of alternatives in the analyzed set changes [8].

In this paper, based on the COMET method's operation, an attempt has been made to determine the effect of overestimation of characteristic values on the results depending on the number of alternatives and criteria. Different levels of overestimation were used to examine and compare the results obtained. The results were then compared using the WS similarity coefficient and the weighted Spearman correlation coefficient to analyze the resulting rankings' correlation.

The rest of the paper is organized as follows. Section 2 presents the preliminaries and main assumptions of the COMET method. Section 3 includes the study case description, where the influence of the overestimation of characteristic values on the received results was examined. Finally, in Section 4 the summary and conclusions from the research are drawn.

II. Preliminaries

A. Weighted Spearman's Rank Coefficient

Weighted Spearman's rank coefficient is defined as (1), where N is a sample size, rank values for both rankings is named as x_i and y_i. In this approach, the positions at the top of both rankings are the most important. The weight of significance is calculated for each alternative. It is the element that determines the main difference to Spearman's rank correlation coefficient, which examines whether the differences appeared and not where they appeared [27].

$$r_w = 1 - \frac{6\sum_{i=1}^{N}(x_i - y_i)^2((N - x_i + 1) + (N - y_i + 1))}{N^4 + N^3 - N^2 - N} \tag{1}$$

B. WS Rank Similarity Coefficient

Rank Similarity Coefficient WS is defined as (2). Unlike r_w, it is an asymmetric measure. The weight of a given comparison is determined based on the significance of the

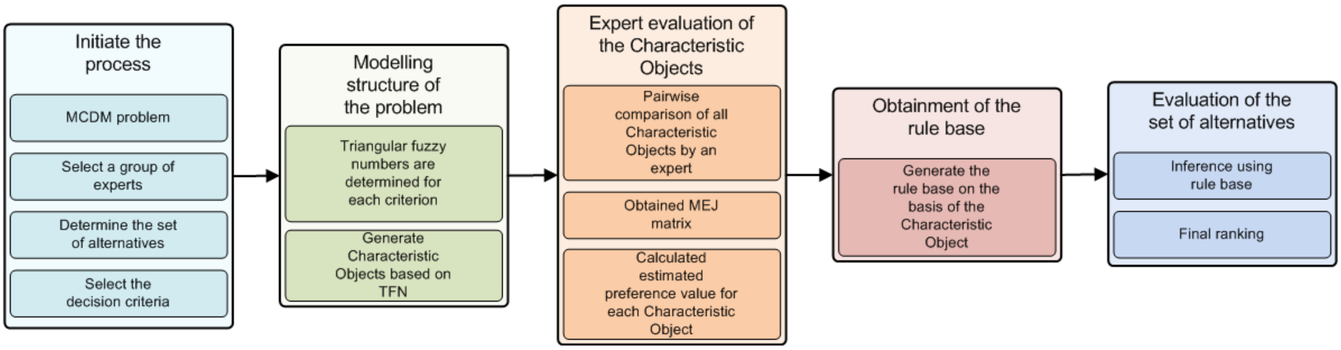

Initiate the process
MCDM problem
Select a group of experts
Determine the set of alternatives
Select the decision criteria
Modelling structure of the problem
Triangular fuzzy numbers are determined for each criterion
Generate Characteristic Objects based on TFN
Expert evaluation of the Characteristic Objects
Pairwise comparison of all Characteristic Objects by an expert
Obtained MEJ matrix
Calculated estimated preference value for each Characteristic Object
Obtainment of the rule base
Generate the rule base on the basis of the Characteristic Object
Evaluation of the set of alternatives
Inference using rule base
Final ranking

Fig. 1. The detailed procedure of the characteristic objects method (COMET).

position in the first ranking, which is used as a reference ranking [28].

$$WS = 1 - \sum_{i=1}^{N} 2^{-x_i} \frac{|x_i - y_i|}{max(|x_i - 1|, |x_i - N|)} \tag{2}$$

C. The Characteristic Objects Method

The Characteristic Objects Method (COMET) is the first method which is completely free of the rank reversal phenomenon [29]. The preferences for the set of alternatives are calculated using the rule base, which is obtained in the process of the pairwise comparison for the Characteristic Objects (COs) [30], [31], [32]. The main assumptions of the COMET method are shortly recalled below following [33]. Additionally, Fig. 1 presents the whole flowchart of the COMET procedure.

Step 1. Define the space of the problem.
An expert determines dimensionality of the problem by selecting number r of criteria, $C_1, C_2, ..., C_r$. Subsequently, the set of fuzzy numbers for each criterion C_i is selected, i.e., $\tilde{C}_{i1}, \tilde{C}_{i2}, ..., \tilde{C}_{ic_i}$. Each fuzzy number determines the value of the membership for a particular linguistic concept for specific crisp values. Therefore it is also useful for variables that are not continuous. In this way, the following result is obtained (3).

$$\begin{array}{l} C_1 = \{\tilde{C}_{11}, \tilde{C}_{12}, ..., \tilde{C}_{1c_1}\} \\ C_2 = \{\tilde{C}_{21}, \tilde{C}_{22}, ..., \tilde{C}_{2c_2}\} \\ \dots\dots\dots\dots\dots\dots\dots\dots \\ C_r = \{\tilde{C}_{r1}, \tilde{C}_{r2}, ..., \tilde{C}_{rc_r}\} \end{array} \tag{3}$$

where $c_1, c_2, ..., c_r$ are numbers of the fuzzy numbers for all criteria.

Step 2. Generate the characteristic objects.
Characteristic objects are objects that define reference points in n-dimensional space. They can be either real or idealized objects that cannot exist [34], [35], [36]. The characteristic objects (CO) are obtained by using the Cartesian product of fuzzy numbers cores for each criterion. As the result, the ordered set of all CO is obtained (4):

$$\begin{array}{l} CO_1 = \{C(\tilde{C}_{11}), C(\tilde{C}_{21}), ..., C(\tilde{C}_{r1})\} \\ CO_2 = \{C(\tilde{C}_{11}), C(\tilde{C}_{21}), ..., C(\tilde{C}_{r2})\} \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ CO_t = \{C(\tilde{C}_{1c_1}), C(\tilde{C}_{2c_2}), ..., C(\tilde{C}_{rc_r})\} \end{array} \tag{4}$$

where t is a number of CO (5):

$$t = \prod_{i=1}^{r} c_i \tag{5}$$

Step 3. Rank the characteristic objects.
The expert determines the Matrix of Expert Judgement (MEJ). It is a result of pairwise comparison of the characteristic objects according to the expert knowledge. The MEJ structure is as follows (6):

$$MEJ = \begin{pmatrix} \alpha_{11} & \alpha_{12} & ... & \alpha_{1t} \\ \alpha_{21} & \alpha_{22} & ... & \alpha_{2t} \\ ... & ... & ... & ... \\ \alpha_{t1} & \alpha_{t2} & ... & \alpha_{tt} \end{pmatrix} \tag{6}$$

where α_{ij} is a result of comparing CO_i and CO_j by the expert [37], [38]. The more preferred characteristic object gets one point and the second object get zero points. If the preferences are balanced, the both objects get half point. It depends solely on the knowledge of the expert and can be presented as (7):

$$\alpha_{ij} = \begin{cases} 0.0, & f_{exp}(CO_i) < f_{exp}(CO_j) \\ 0.5, & f_{exp}(CO_i) = f_{exp}(CO_j) \\ 1.0, & f_{exp}(CO_i) > f_{exp}(CO_j) \end{cases} \tag{7}$$

where f_{exp} is an expert mental judgement function. Afterwards, the vertical vector of the Summed Judgements (SJ) is obtained as follows (8):

$$SJ_i = \sum_{j=1}^{t} \alpha_{ij} \tag{8}$$

The number of query is equal to $p = \frac{t(t-1)}{2}$ because for each element α_{ij} we can observe that $\alpha_{ji} = 1 - \alpha_{ij}$. In the last step, an approximate value of preference P_i is assigned to each characteristic object using Algorithm 1. As a result, vector P is obtained, where i-th row contains the approximate value of preference for CO_i.

Algorithm 1

```
k = length(unique(SJ));
P = zeros(t, 1);
for i = 1:k
    ind = find(SJ == max(SJ));
    p(ind) = (k - i)/(k - 1);
    SJ(ind) = 0;
end
```

Step 4. The rule base.
Each characteristic object is converted into a fuzzy rule, where the degree of belonging to particular criteria is a premise for activating conclusions in the form of P_i. Each characteristic object and value of preference is converted to a fuzzy rule (for more details see [39]). n this way, the complete fuzzy rule base that approximates the expert mental judgement function $f_{exp}(CO_i)$ is obtained.

Step 5. Inference and final ranking.
The each one alternative A_i is a set of crisp numbers a_{ri} corresponding to criteria $C_1, C_2, ..., C_r$. It can be presented as follows (9):

$$A_i = \{a_{1i}, a_{2i}, ..., a_{ri}\} \tag{9}$$

Each alternative activates the specified number of fuzzy rules, where for each one the fulfilment degree of the complex conjunctive premise is determined. Fulfilment degrees of all activated rules are summed up to one. The preference of alternative is computed as the sum of products of all activated rules, their fulfilment degrees, and their values of the preference. The final ranking of alternatives is obtained by sorting the preference of alternatives, where one is the best result, and zero is the worst. More details can be found in [40], [41].

III. Case Study

To determine the impact of overestimating the characteristic values in the application of the COMET method, it was decided to carry out two research cases, in which changes in the obtained rankings were observed. In the first case study different numbers of alternatives ([5, 10, 15, 20, 30]) were taken into account, with the number of criteria equal to 4. The second case study involved changes in the number of criteria ([4, 6, 8]), with 10 alternatives in the considered set.

Additionally, in both research cases, the types of criteria were determined by dividing the number of criteria in half and assigning the profit type to one of them and the cost type to the other. Moreover, the established levels of overestimation of characteristic values were described by the range [0.05, 0.10, 0.15, 0.20, 0.25, 0.30]. It is also worth mentioning that the Characteristic Objects values were defined as [0, 0.5, 1]. The overestimation level was subtracted and added for the lower and upper limits of the COs, respectively. The obtained alternatives' preference rankings were then compared using the WS similarity coefficient and the weighted Spearman correlation coefficient.

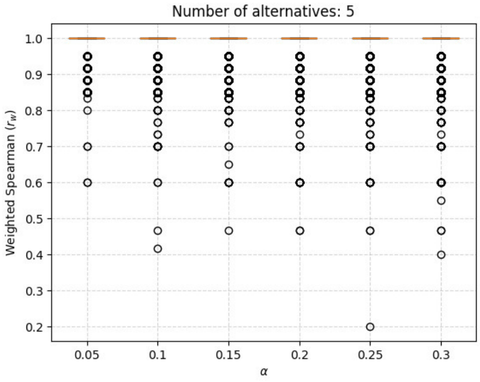


Fig. 2. Distribution of r_w similarity coefficient for rankings with five alternatives and four criteria.

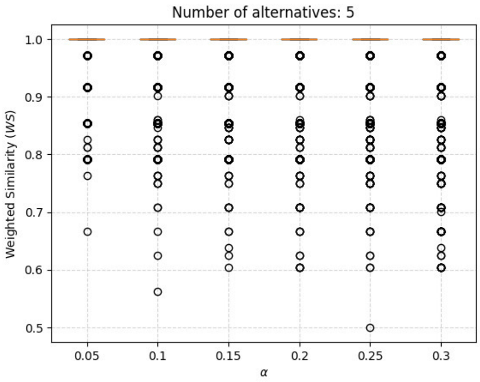


Fig. 3. Distribution of WS similarity coefficient for rankings with five alternatives and four criteria.

Fig. 2 and 3 present a visualization of the ranking values obtained for a matrix size of 5 alternatives and 4 criteria for both similarity coefficients. In both cases, it can be seen that the least divergent values were obtained in the case when the overestimation level was 0.05. In the WS coefficient case, the correlation values were less differentiated. They mainly oscillated in the range [0.6, 1.0], while for the Spearman coefficient, the values were more diverse, where the interval settled in the range [0.2, 1.0]. It can also be noted that the change in the overestimation of values in the examined interval [0.15, 0.30] did not significantly affect the differences between the rankings.

A test case with 15 alternatives and 4 criteria in the decision matrix showed that when the overestimation value was increased, the rankings' similarity decreased slightly for both similarity coefficients used. The values of the weighted Spearman coefficient returned higher similarity than the WS similarity coefficient. In the cases analyzed for the different overestimation values, the rankings showed a high similarity of above 0.92 for the Spearman coefficient and above 0.825 for the WS coefficient. The visualization is shown in Fig. 4 and 5.

In turn, Fig. 6 and 7 show the correlation values obtained

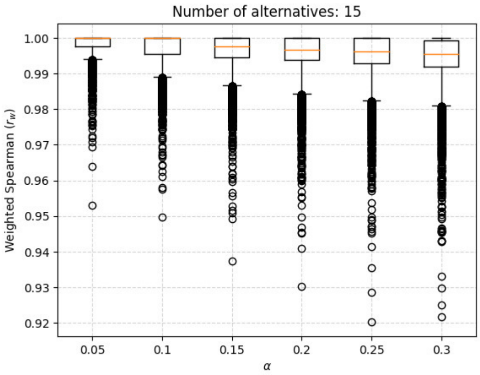


Fig. 4. Distribution of r_w similarity coefficient for rankings with fifteen alternatives and four criteria.

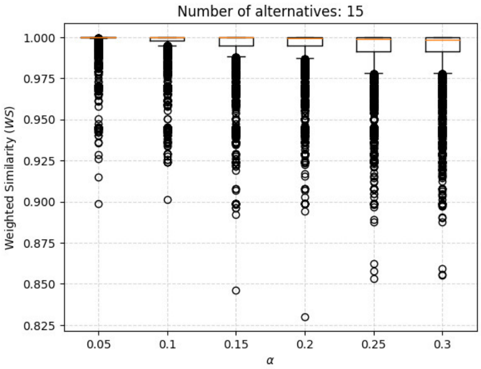


Fig. 5. Distribution of WS similarity coefficient for rankings with fifteen alternatives and four criteria.

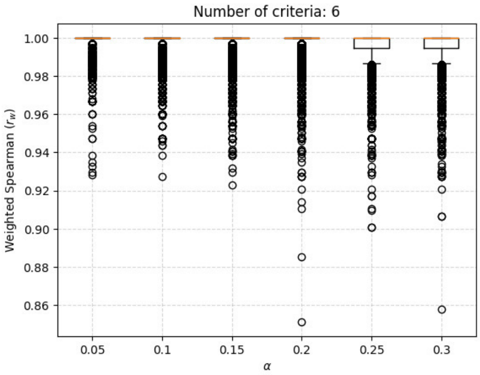


Fig. 6. Distribution of r_w coefficient for rankings with ten alternatives and six criteria.

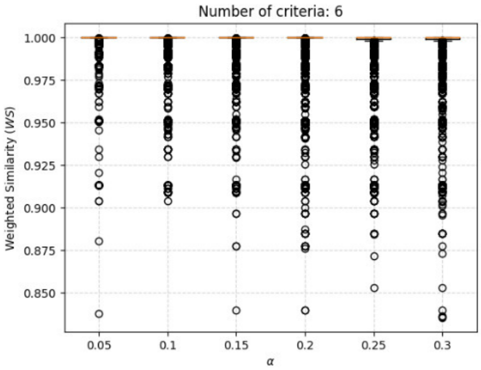


Fig. 7. Distribution of WS similarity coefficient for rankings with ten alternatives and six criteria.

for the defined range of overestimation values and the decision matrix for 10 alternatives and 6 criteria. The returned correlations were similar and guaranteed values in the range [0.85, 1.0]. It is worth noting that increasing the overestimation value again affected obtaining more significant variation in the rankings' similarity.

More significant overestimation had a negative effect on the obtained rankings correlation values and resulted in greater diversity. Additionally, it was noted that an increase in the number of criteria has a positive impact on the recorded similarity of the rankings.

IV. Conclusions

Determining the impact of decisions made by the expert when using multi-criteria methods is an important element in the process of evaluating alternatives. The data defined by the expert, based on his knowledge and experience, can sometimes be overestimated, and it can directly impact the results achieved.

For this purpose, it was decided to use the COMET method, in which a numerical interval containing three values must be specified to define the Characteristic Objects. Values that change the baseline boundary limits were defined, which allowed the study of overestimating the final rankings. Two test cases were conducted in which changes in the number of alternatives with a constant number of criteria and changes in the number of criteria with a constant number of alternatives were investigated. The resulting rankings were then compared using two selected similarity coefficients. It was observed that a higher number of alternatives positively affects the correlation strength of the rankings. On the other hand, increasing the overestimation of the boundary values decreases the obtained results' similarity.

For further directions, it is worth considering how the overestimation affects the results when fuzzy extensions are involved in the COMET method application. Moreover, it can be determined whether the overestimation occurring in other MCDM methods can affect the resulting rankings.

Acknowledgment

The work was supported by the National Science Centre, Decision number UMO-2018/29/B/HS4/02725 (W.S.).

References

[1] J. J. Jassbi, R. A. Ribeiro, and L. R. Varela, "Dynamic mcdm with future knowledge for supplier selection," *Journal of Decision Systems*, vol. 23, no. 3, pp. 232–248, 2014.

[2] I. Vinogradova, V. Podvezko, and E. K. Zavadskas, "The recalculation of the weights of criteria in mcdm methods using the bayes approach," *Symmetry*, vol. 10, no. 6, p. 205, 2018.

[3] D. S. Pamučar, D. Božanić, and A. Ranđelović, "Multi-criteria decision making: An example of sensitivity analysis," *Serbian journal of management*, vol. 12, no. 1, pp. 1–27, 2017.

[4] S. Opricovic and G.-H. Tzeng, "Compromise solution by mcdm methods: A comparative analysis of vikor and topsis," *European journal of operational research*, vol. 156, no. 2, pp. 445–455, 2004.

[5] L. Ustinovichius, E. Zavadkas, and V. Podvezko, "Application of a quantitative multiple criteria decision making (mcdm-1) approach to the analysis of investments in construction," *Control and cybernetics*, vol. 36, no. 1, p. 251, 2007.

[6] B. Kizielewicz, J. Więckowski, A. Shekhovtsov, E. Ziemba, J. Wątróbski, and W. Sałabun, "Input data preprocessing for the mcdm model: Copras method case study," 2021.

[7] S. Opricovic and G.-H. Tzeng, "Extended vikor method in comparison with outranking methods," *European journal of operational research*, vol. 178, no. 2, pp. 514–529, 2007.

[8] W. Sałabun, P. Ziemba, and J. Wątróbski, "The rank reversals paradox in management decisions: The comparison of the ahp and comet methods," in *International Conference on Intelligent Decision Technologies*. Springer, 2016, pp. 181–191.

[9] A. Krylovas, E. K. Zavadskas, N. Kosareva, and S. Dadelo, "New kemira method for determining criteria priority and weights in solving mcdm problem," *International Journal of Information Technology & Decision Making*, vol. 13, no. 06, pp. 1119–1133, 2014.

[10] D. Pamučar, Ž. Stević, and S. Sremac, "A new model for determining weight coefficients of criteria in mcdm models: Full consistency method (fucom)," *Symmetry*, vol. 10, no. 9, p. 393, 2018.

[11] W. Sałabun, A. Karczmarczyk, J. Wątróbski, and J. Jankowski, "Handling data uncertainty in decision making with comet," in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2018, pp. 1478–1484.

[12] B. Kizielewicz and L. Dobryakova, "Mcda based approach to sports players' evaluation under incomplete knowledge," *Procedia Computer Science*, vol. 176, pp. 3524–3535, 2020.

[13] Y.-C. Chen, H.-P. Lien, and G.-H. Tzeng, "Measures and evaluation for environment watershed plans using a novel hybrid mcdm model," *Expert systems with applications*, vol. 37, no. 2, pp. 926–938, 2010.

[14] Y.-C. Lee, P.-H. Chung, and J. Z. Shyu, "Performance evaluation of medical device manufacturers using a hybrid fuzzy mcdm," 2017.

[15] O. Parkash and R. Kumar, "Modified fuzzy divergence measure and its applications to medical diagnosis and mcdm," *Risk and Decision Analysis*, vol. 6, no. 3, pp. 231–237, 2017.

[16] Ž. Stević, D. Pamučar, A. Puška, and P. Chatterjee, "Sustainable supplier selection in healthcare industries using a new mcdm method: Measurement of alternatives and ranking according to compromise solution (marcos)," *Computers & Industrial Engineering*, vol. 140, p. 106231, 2020.

[17] J. Roy, K. Adhikary, S. Kar, and D. Pamucar, "A rough strength relational dematel model for analysing the key success factors of hospital service quality," *Decision Making: Applications in Management and Engineering*, vol. 1, no. 1, pp. 121–142, 2018.

[18] V. Y. Chen, H.-P. Lien, C.-H. Liu, J. J. Liou, G.-H. Tzeng, and L.-S. Yang, "Fuzzy MCDM approach for selecting the best environment-watershed plan," *Applied soft computing*, vol. 11, no. 1, pp. 265–275, 2011.

[19] A. Shekhovtsov, V. Kozlov, V. Nosov, and W. Sałabun, "Efficiency of methods for determining the relevance of criteria in sustainable transport problems: A comparative case study," *Sustainability*, vol. 12, no. 19, p. 7915, 2020.

[20] Ž. Stević, D. Pamučar, M. Vasiljević, G. Stojić, and S. Korica, "Novel integrated multi-criteria model for supplier selection: Case study construction company," *Symmetry*, vol. 9, no. 11, p. 279, 2017.

[21] M.-T. Lu, C.-C. Hsu, J. J. Liou, and H.-W. Lo, "A hybrid mcdm and sustainability-balanced scorecard model to establish sustainable performance evaluation for international airports," *Journal of Air Transport Management*, vol. 71, pp. 9–19, 2018.

[22] M. Nassereddine and H. Eskandari, "An integrated mcdm approach to evaluate public transportation systems in tehran," *Transportation Research Part A: Policy and Practice*, vol. 106, pp. 427–439, 2017.

[23] B. Kizielewicz, J. Więckowski, A. Shekhovtsov, J. Wątróbski, R. Depczyński, and W. Sałabun, "Study towards the time-based mcda ranking analysis–a supplier selection case study," *Facta Universitatis, Series: Mechanical Engineering*, 2021.

[24] S. Faizi, W. Sałabun, S. Ullah, T. Rashid, and J. Więckowski, "A new method to support decision-making in an uncertain environment based on normalized interval-valued triangular fuzzy numbers and comet technique," *Symmetry*, vol. 12, no. 4, p. 516, 2020.

[25] W. Sałabun, "Reduction in the number of comparisons required to create matrix of expert judgment in the comet method," *Management and Production Engineering Review*, vol. 5, 2014.

[26] B. Kizielewicz and J. Kołodziejczyk, "Effects of the selection of characteristic values on the accuracy of results in the comet method," *Procedia Computer Science*, vol. 176, pp. 3581–3590, 2020.

[27] W. Sałabun, J. Wątróbski, and A. Shekhovtsov, "Are MCDA methods benchmarkable? a comparative study of TOPSIS, VIKOR, COPRAS, and PROMETHEE II methods," *Symmetry*, vol. 12, no. 9, p. 1549, 2020.

[28] W. Sałabun and K. Urbaniak, "A new coefficient of rankings similarity in decision-making problems," in *International Conference on Computational Science*. Springer, 2020, pp. 632–645.

[29] W. Sałabun, A. Piegat, J. Wątróbski, A. Karczmarczyk, and J. Jankowski, "The comet method: The first mcda method completely resistant to rank reversal paradox," *European Working Group Series*, vol. 3.

[30] K. Palczewski and W. Sałabun, "Identification of the football teams assessment model using the comet method," *Procedia Computer Science*, vol. 159, pp. 2491–2501, 2019.

[31] W. Sałabun, A. Shekhovtsov, and B. Kizielewicz, "A new consistency coefficient in the multi-criteria decision analysis domain," in *International Conference on Computational Science*. Springer, 2021, pp. 715–727.

[32] B. Kizielewicz and Z. Szyjewski, "Handling economic perspective in multicriteria model-renewable energy resources case study," *Procedia Computer Science*, vol. 176, pp. 3555–3562, 2020.

[33] W. Sałabun, A. Shekhovtsov, D. Pamučar, J. Wątróbski, B. Kizielewicz, J. Więckowski, D. Bozanić, K. Urbaniak, and B. Nyczaj, "A fuzzy inference system for players evaluation in multi-player sports: The football study case," *Symmetry*, vol. 12, no. 12, p. 2029, 2020.

[34] B. Kizielewicz and W. Sałabun, "A new approach to identifying a multi-criteria decision model based on stochastic optimization techniques," *Symmetry*, vol. 12, no. 9, p. 1551, 2020.

[35] J. Więckowski, B. Kizielewicz, and J. Kołodziejczyk, "Finding an approximate global optimum of characteristic objects preferences by using simulated annealing," in *International Conference on Intelligent Decision Technologies*. Springer, 2020, pp. 365–375.

[36] ——, "The search of the optimal preference values of the characteristic objects by using particle swarm optimization in the uncertain environment," in *International Conference on Intelligent Decision Technologies*. Springer, 2020, pp. 353–363.

[37] B. Kizielewicz, A. Shekhovtsov, and W. Sałabun, "A new approach to eliminate rank reversal in the mcda problems," in *International Conference on Computational Science*. Springer, 2021, pp. 338–351.

[38] B. Kizielewicz and L. Dobryakova, "How to choose the optimal single-track vehicle to move in the city? electric scooters study case," *Procedia Computer Science*, vol. 176, pp. 2243–2253, 2020.

[39] W. Sałabun, "The characteristic objects method: A new distance-based approach to multicriteria decision-making problems," *Journal of Multi-Criteria Decision Analysis*, vol. 22, no. 1-2, pp. 37–50, 2015.

[40] A. Piegat and W. Sałabun, "Identification of a multicriteria decision-making model using the characteristic objects method," *Applied Computational Intelligence and Soft Computing*, vol. 2014, 2014.

[41] W. Sałabun and A. Piegat, "Comparative analysis of MCDM methods for the assessment of mortality in patients with acute coronary syndrome," *Artificial Intelligence Review*, vol. 48, no. 4, pp. 557–571, 2017.

27th Conference on Knowledge Acquisition and Management

KNOWLEDGE management is a large multidisciplinary field having its roots in Management and Artificial Intelligence. Activity of an extended organization should be supported by an organized and optimized flow of knowledge to effectively help all participants in their work.

We have the pleasure to invite you to contribute to and to participate in the conference “Knowledge Acquisition and Management". The predecessor of the KAM conference has been organized for the first time in 1992, as a venue for scientists and practitioners to address different aspects of usage of advanced information technologies in management, with focus on intelligent techniques and knowledge management. In 2003 the conference changed somewhat its focus and was organized for the first under its current name. Furthermore, the KAM conference became an international event, with participants from around the world. In 2012 we’ve joined to Federated Conference on Computer Science and Systems becoming one of the oldest event.

The aim of this event is to create possibility of presenting and discussing approaches, techniques and tools in the knowledge acquisition and other knowledge management areas with focus on contribution of artificial intelligence for improvement of human-machine intelligence and face the challenges of this century. We expect that the conference&workshop will enable exchange of information and experiences, and delve into current trends of methodological, technological and implementation aspects of knowledge management processes.

Topics

- Knowledge discovery from databases and data warehouses
- Methods and tools for knowledge acquisition
- New emerging technologies for management
- Organizing the knowledge centers and knowledge distribution
- Knowledge creation and validation
- Knowledge dynamics and machine learning
- Distance learning and knowledge sharing
- Knowledge representation models
- Management of enterprise knowledge versus personal knowledge
- Knowledge managers and workers
- Knowledge coaching and diffusion
- Knowledge engineering and software engineering
- Managerial knowledge evolution with focus on managing of best practice and cooperative activities
- Knowledge grid and social networks
- Knowledge management for design, innovation and eco-innovation process
- Business Intelligence environment for supporting knowledge management
- Knowledge management in virtual advisors and training
- Management of the innovation and eco-innovation process
- Human-machine interfaces and knowledge visualization

Technical Session Chairs

- **Hauke, Krzysztof,** Wroclaw University of Economics, Poland
- **Nycz, Malgorzata,** Wroclaw University of Economics, Poland
- **Owoc, Mieczyslaw,** Wroclaw University of Economics, Poland
- **Pondel, Maciej,** Wroclaw University of Economics, Poland

Program Committee

- **Abramowicz, Witold,** Poznan University of Economics, Poland
- **Andres, Frederic,** National Institute of Informatics, Tokyo, Japan
- **Bodyanskiy, Yevgeniy,** Kharkiv National University of Radio Electronics, Ukraine
- **Chmielarz, Witold,** Warsaw University, Poland
- **Christozov, Dimitar,** American University in Bulgaria, Bulgaria
- **Jan, Vanthienen,** Katholike Universiteit Leuven, Belgium
- **Mercier-Laurent, Eunika,** University Jean Moulin Lyon3, France
- **Sobińska, Małgorzata,** Wroclaw University of Economics, Poland
- **Surma, Jerzy,** Warsaw School of Economics, Poland and University of Massachusetts Lowell, United States
- **Vasiliev, Julian,** University of Economics in Varna, Bulgaria
- **Zhu, Yungang,** College of Computer Science and Technology, Jilin University, China

DOI: 10.15439/2021F118
ISSN 2300-5963 ACSIS, Vol. 25

Using Word Embeddings for Italian Crime News Categorization

Giovanni Bonisoli, Federica Rollo, Laura Po
Enzo Ferrari Engineering Department
University of Modena and Reggio Emilia
Italy
Email: 204058@studenti.unimore.it, federica.rollo@unimore.it, laura.po@unimore.it

***Abstract*—Several studies have shown that the use of embeddings improves outcomes in many Natural Language Processing (NLP) activities, including text categorization. This paper focuses on how word embeddings can be used on newspaper articles related to crimes. The scope is the categorization of the news articles based on the type of crime they report. We compare different Word2Vec models and methods to obtain word embeddings. Then, we exploit both supervised and unsupervised Machine Learning categorization algorithms. Experiments were conducted on an Italian dataset of 15,361 crime news articles showing very promising results.**

I. Introduction

THE categorization of news articles consists of understanding the topic of the articles and associate each of them to a category. In the case of news articles related to crimes, the scope is to identify the type of crime (*crime categorization*). This task is important for many reasons. The first one is the need to create statistics on the type of events. Indeed, categorization allows understanding how often and where a certain type of crime occurs [1]. Secondly, categorization enables for further processing that are in the scope of crime analysis. From each news article, it is possible to retrieve detailed information about the event it reports: the place, the thief, the victim [2]. If we know the type of crime, we can also retrieve information specific to that crime type, e.g. the stolen items in a theft. Analyzing crime news articles allows also to study how exposure to crime news articles content is associated with perceived social trust [3]. Moreover, Machine Learning approaches can help crime analysts to identify the connected events and to generate alerts and predictions that lead to better decision-making and optimized actions [4].

Several studies concerning crime analysis exploit news articles [5–7]. In most cases, due to the lack of official data, newspapers are a valuable source of authentic and timely information [8]. Detailed information can be extracted through the application of Natural Language Processing (NLP) techniques.

According to the use case, the scope of assigning a news article to a crime category can be addressed following several approaches, such as text classification, community or topic detection [9–12].

In text classification, it seems appealing to enhance word representations with ad-hoc embeddings that encode task-specific information [13]. Word embedding is a continuous vector representation of words that encodes the meaning of the word, such that the words that are closer in the vector space are expected to be similar in meaning. There are different machine learning algorithms that can be trained to derive these vectors, such Word2Vec [14], FastText [15], Glove [16]. The use of word embeddings as additional features improves the performance in many NLP tasks, including text classification [17–21]. The authors of [22] and [23] suggested different kinds of features to derive from word embeddings and tested them as features in the classification task.

In this paper, we introduce an approach to perform crime categorization on Italian news articles. The work is inspired by the previous approaches that use word embeddings to classify texts about other topics [22, 23].

The paper is organized as follows. The general approach is described in Section II. In the following, we describe our dataset (Section III) and three models used to generate word embeddings (Section IV). Section V details the experimental results of crime categorization, which is performed using supervised and unsupervised techniques, and shows empirical evidence of high accuracy. Section VI is dedicated to conclusions.

II. Proposed Approach

The general procedure consists of the use of word embeddings to obtain features to be given as input to a categorization algorithm. To obtain the feature vector of each news article, its text is pre-processed by executing:

1) *Tokenization*, which returns the list of the words that are present in the text.
2) *Stop word removal*, a commonly used technique before performing NLP tasks since stop words occur a lot of times in texts and do not provide any relevant information. The result is a list of the most relevant words that are present in the text.
3) *Lemmatization*, the process of deriving the lemma of a word. Every word in the list is replaced by its lemma.

At the end of these phases, the final result is a list of meaningful words for every news article.

Then, using a trained word embedding model, we get a lookup table where each word is replaced by its corresponding word vector (word embedding). If a word in the text is not found in the vocabulary of the model, it is simply discarded from the list without any replacement. As the authors of [22] suggest, for each news article two vector representations can be extracted by using the word embeddings:

- the simple average of the word vectors,
- the average of the word vectors weighted by the TF-IDF score of each word computed on the text of the news articles in the dataset. This representation gives more importance to those vectors that are related to words with a high frequency in the text of a news article and a low frequency in the others.

Each type of vector representation can be calculated on the non-lemmatized list of words obtained at the second step of the pre-processing, or on the lemmatized list obtained at the third step. In this way, four feature vectors can be obtained for each news article: simple average without lemmatization, simple average with lemmatization, TF-IDF weighted average without lemmatization, and TF-IDF weighted average with lemmatization. Then, it is possible to compare the results and evaluate the impact of lemmatization and the choice of the type of average on the downstream task. Figure 1 illustrates the entire pre-process. The obtained word vectors are the input data for any categorization algorithm. As described in the following sections, we use Word2Vec as a word embedding model and perform categorization through both supervised and unsupervised algorithms.

III. Italian Crime News Dataset

The experiments are conducted using an Italian dataset of crime news articles. The information about the news articles is collected by the Crime Ingestion App [8], a Java application that aims at extracting, geolocalizing and deduplicating crime-related news articles from two online newspapers of the province of Modena in Italy (“ModenaToday”[1] and “Gazzetta di Modena”[2]).

The data extracted from the newspapers include the *URL* of the web page containing the news article, the *title* of the news article, the *sub-title*, the *text*, the information related to the place where the crime occurred (*municipality*, *area*, and *address*), the *publication_datetime* that is the date and the time of publication of the news article, and the *event_datetime* that refers to the date of crime event. Part of these data is automatically extracted from the web page of the news articles, the other ones are identified by applying NLP techniques to the text of the news articles. Besides, the newspapers we consider already classify news articles according to the crime type (this classification is done manually by the journalist, author of the news articles). Each news article is assigned to a specific crime category. The total number of categories is 13: “furto” (theft), “rapina” (robbery), “omicidio” (murder), “violenza sessuale” (sexual violence), “maltrattamento” (mis-treatment), “aggressione” (aggression), “spaccio” (illegal sale, most commonly used to refer to drug trufficking), “droga” (drug dealing), “truffa” (scam), “frode” (fraud), “riciclaggio” (money laundering), “evasione” (evasion), and “sequestro” (kidnapping).

The current dataset contains 15,361 news articles published in the two selected newspapers from 2011 to now (approximately 9 years).

IV. Word2Vec Models

Word2Vec is based on a shallow neural network whose input data are generated by a window sliding on the text of the training corpus. This window selects a context within which it chooses a target to obscure and predict based on the rest of the selected context. Through this “fake task” internal parameters of the network are learned which constitute word embedding, the real objective of training. Three Word2Vec models are chosen for our experiments:

M1 a pre-trained model [24], whose dimension is 300. The dataset used to train Word2Vec was obtained exploiting the information extracted from a dump of Wikipedia, the main categories of Italian Google News and some anonymized chats between users and the customer care chatbot Laila.[3] The dataset (composed of 2.6 GB of raw text) includes 17,305,401 sentences and 421,829,960 words.

M2 A Skip-Gram model trained from scratch on the crime news articles of our dataset for 30 epochs (*window_size=10, min_count=20, negative_sampling=20, embedding_dim=300*).

M3 A Skip-Gram model which has been trained on the crime news articles of our dataset for 5 epochs, starting from the embeddings of M1 (*window_size=10, min_count=20, negative_sampling=20, embedding_dim=300*).

V. Crime Categorization

After obtaining the vector representations of each news article in the dataset, several algorithms can be used to identify the category each news article belongs to. Both supervised and unsupervised techniques can be taken into account. In the following, Section V-A presents our tests with supervised text categorization algorithms, while Section V-B discusses some experiments with unsupervised methods.

A. Supervised Text Categorization

Supervised text categorization algorithms predict the topic of a document within a predefined set of categories, named labels. In this case, the labels are the crime categories listed in Section III and the documents are the texts of the crime news articles.

The embeddings obtained by the three Word2Vec models described in Section IV are tested for categorization. Different supervised machine learning algorithms have been exploited as suggested in [25]. Around 65% of the articles in the dataset is used as the training set (10,138 articles), while the remaining is used as the test set (5,223 articles). Both sets contain articles

[1] https://www.modenatoday.it/

[2] https://gazzettadimodena.gelocal.it/modena

[3] https://www.laila.tech/

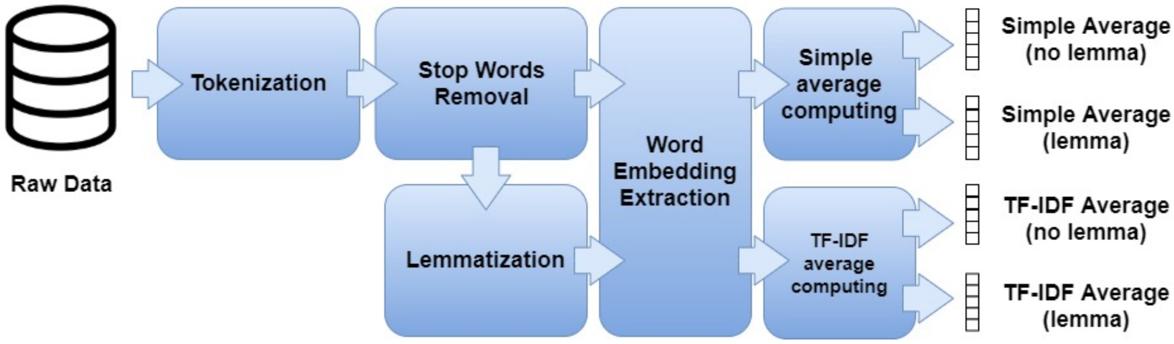


Fig. 1. Feature extraction.

from both newspapers. Table I shows the number of articles for each category that are included in each set. As can be noticed, there is a considerable imbalance of the categories in both sets. The dominant category is “theft”.

TABLE I
THE NUMBER OF NEWS ARTICLES IN THE TRAINING AND TEST SETS FOR EACH CATEGORY.

Category	*Training Set*	*Test Set*
Theft	6231	3212
Drug dealing	1020	541
Illegal sale	632	344
Aggression	513	258
Robbery	508	273
Scam	368	189
Mistreatment	161	79
Murder	153	76
Evasion	149	83
Kidnapping	139	71
Money laundering	67	43
Sexual violence	61	30
Fraud	20	17
Total	10138	5223

Table II, III and IV show the results of 15 supervised algorithms trained on the feature vectors obtained by the embeddings of M1, M2, M3 respectively. In the tables, the first column contains the name of the categorization algorithm employed, in the other columns there are the values of accuracy obtained by using simple average or TF-IDF weighted average and including or excluding lemmatization. As can be seen, the absence of lemmatization has little influence on accuracy both in the simple average and the TF-IDF weighted average for all the algorithms and models. Instead, there is a substantial difference when passing from the simple average to the TF-IDF weighted average for M1. The latter brings a notable improvement in performance in most of the algorithms. As shown in Table II, four algorithms have accuracy greater than 0.75: SGD (L2 norm regularization, Hinge loss), SVC (RBF Kernel, gamma=’scale’), Linear SVC (C=1.0), and XGBboost. All the accuracy values are lower than 0.80. In few cases, accuracy is higher than 0.75. Also, some algorithms achieved a very low accuracy (0.04-0.38). Since “theft” is the most present category, the overall accuracy depends a lot on the accuracy reported in this category. Therefore, low values of the overall accuracy corresponds to low accuracy in the category “theft”. Besides, there are some cases where medium-high overall accuracy (0.46-0.64) corresponds to a high accuracy on the category “theft” while the accuracy on the other categories is very low or zero.

M2 outperforms M1 in terms of accuracy. As shown in Table III, some values of accuracy are greater than 80%. This is probably due to the fact that the feature vectors are derived from embeddings learned on the same documents (M2 is indeed trained on the crime news). This makes certain words more discriminative for certain contexts, and therefore, for certain crime categories. In M2, there is no improvement when passing from the use of the simple average to the TF-IDF weighted average. There are four algorithms with at least one accuracy value greater than 0.80; they are the same best algorithms retrieved with M1: SGD (L2 norm regularization, Hinge loss), SVC (RBF kernel, C=1.0, gamma=‘scale’), Linear SVC (C=1.0), and XGBboost.

Table IV shows the results of the supervised categorization using the feature vectors obtained by the embeddings of M3. The performances are comparable to those obtained in Table III. Besides, also in this case, we do not find any significant difference between the use of simple average and TF-IDF weighted average. This is probably due to the fact that the embedding of M3 are obtained retraining M1 on our dataset. The embeddings of M1 are trained on a dataset that largely includes news articles, thus it contains contexts very similar to the ones of our dataset. Therefore, retraining them on our dataset probably led to embeddings similar to the ones of M2.

Table V shows in detail the results of the best algorithm (Linear SVC) using the embeddings of M3 in the supervised categorization for each category. The third column indicates the number of news articles in the test set for each category. The values of precision and recall show that the algorithm suffers from the imbalance of the training set. The less the category is present in the dataset, the more the recall (sometimes also the precision) decreases.

After some analysis, we discovered that the annotation of the news articles published in “Gazzetta di Modena” is not so accurate, so these tests on categorization are “dirty”. Then, we decide to perform the test again by using the embeddings of

TABLE II
ACCURACY OF THE APPLICATION OF DIFFERENT CATEGORIZATION ALGORITHMS ON THE EMBEDDINGS DERIVED FROM M1.

Algorithm	*Simple average (no lemma)*	*Simple average (lemma)*	*TF-IDF average (no lemma)*	*TF-IDF average (lemma)*
SGD (L2 norm, Hinge loss)	0.62	0.61	**0.77**	0.74
SGD (L1 norm, Perceptron)	0.04	0.22	0.72	0.74
SVC (RBF kernel, C=1.0, gamma='scale')	0.62	0.62	**0.76**	**0.75**
Linear SVC (C=1.0)	0.62	0.62	**0.77**	**0.77**
GaussianNB	0.38	0.32	0.48	0.45
BernoulliNB	0.62	0.62	0.57	0.58
K-nearest-neighbour (k=1)	0.50	0.49	0.68	0.68
K-nearest-neighbour (k=3)	0.59	0.59	0.72	0.71
K-nearest-neighbour (k=5)	0.61	0.60	0.73	0.73
Decision Tree	0.46	0.46	0.56	0.55
Random Forest Classifier (n=100)	0.64	0.63	0.69	0.68
Adaboost (DecisionTree)	-	0.60	0.61	0.58
Bagging (DecisionTree)	0.61	0.60	0.68	0.67
Bagging (KNN(n=5))	0.60	0.60	0.74	0.73
XGBboost	0.63	0.64	**0.75**	**0.75**

TABLE III
ACCURACY OF THE APPLICATION OF DIFFERENT CATEGORIZATION ALGORITHMS ON THE EMBEDDINGS DERIVED FROM M2.

Algorithm	*Simple average (no lemma)*	*Simple average (lemma)*	*TF-IDF average (no lemma)*	*TF-IDF average (lemma)*
SGD (L2 norm, Hinge loss)	**0.82**	**0.82**	0.78	0.78
SGD (L1 norm, Perceptron)	0.79	0.79	0.77	0.75
SVC (RBF kernel, C=1.0, gamma='scale')	**0.83**	**0.83**	**0.83**	**0.83**
Linear SVC (C=1.0)	**0.82**	**0.83**	0.79	0.79
GaussianNB	0.59	0.62	0.61	0.63
BernoulliNB	0.55	0.62	0.57	0.60
K-nearest-neighbour (k=1)	0.59	0.75	0.76	0.76
K-nearest-neighbour (k=3)	0.78	0.78	0.78	0.76
K-nearest-neighbour (k=5)	0.79	0.79	0.79	0.73
Decision Tree	0.66	0.68	0.65	0.67
Random Forest Classifier (n=100)	0.76	0.77	0.77	0.77
Adaboost (DecisionTree)	-	0.62	0.64	0.60
Bagging (DecisionTree)	0.75	0.76	0.75	0.76
Bagging (KNN(n=5))	0.79	0.79	0.79	0.79
XGBboost	**0.82**	**0.82**	**0.82**	**0.81**

TABLE IV
ACCURACY OF THE APPLICATION OF DIFFERENT CATEGORIZATION ALGORITHMS ON THE EMBEDDINGS DERIVED FROM M3.

Algorithm	*Simple average (no lemma)*	*Simple average (lemma)*	*TF-IDF average (no lemma)*	*TF-IDF average (lemma)*
SGD (L2 norm, Hinge loss)	**0.82**	**0.81**	**0.80**	**0.81**
SGD (L1 norm, Perceptron)	0.76	0.75	0.77	0.75
SVC (RBF kernel, C=1.0, gamma='scale')	**0.82**	**0.84**	**0.81**	**0.81**
Linear SVC (C=1.0)	**0.83**	**0.84**	**0.81**	**0.82**
GaussianNB	0.64	0.64	0.63	0.62
BernoulliNB	0.62	0.65	0.62	0.65
K-nearest-neighbour (k=1)	0.76	0.76	0.75	0.75
K-nearest-neighbour (k=3)	0.78	0.78	0.78	0.77
K-nearest-neighbour (k=5)	0.79	0.79	0.79	0.77
Decision Tree	0.65	0.66	0.79	0.77
Random Forest Classifier (n=100)	0.76	0.76	0.75	0.76
Adaboost (DecisionTree)	0.76	-	0.75	-
Bagging (DecisionTree)	0.75	-	0.74	-
Bagging (KNN(n=5))	0.79	-	0.79	-
XGBboost	**0.81**	-	**0.80**	-

TABLE V
PRECISION AND RECALL OF LINEAR SVC ON CATEGORIZATION USING THE EMBEDDINGS OF M3.

Category	*news articles*	*Simple average (no lemma)*		*Simple average (lemma)*		*TF-IDF average (no lemma)*		*TF-IDF average (lemma)*	
		precision	*recall*	*precision*	*recall*	*precision*	*recall*	*precision*	*recall*
Theft	6267	0.89	0.95	0.89	0.95	0.88	0.94	0.89	0.94
Drug Dealing	1018	0.71	0.67	0.68	0.74	0.69	0.63	0.66	0.69
Illegal sale	636	0.66	0.55	0.76	0.52	0.62	0.55	0.70	0.51
Aggression	529	0.69	0.69	0.70	0.69	0.71	0.66	0.68	0.67
Robbery	516	0.79	0.59	0.81	0.67	0.76	0.59	0.73	0.68
Scam	376	0.72	0.73	0.73	0.75	0.74	0.71	0.72	0.70
Mistreatment	170	0.69	0.60	0.69	0.62	0.68	0.59	0.66	0.59
Murder	159	0.69	0.73	0.75	0.61	0.59	0.61	0.70	0.53
Evasion	164	0.80	0.52	0.83	0.53	0.62	0.58	0.68	0.54
Kidnapping	134	0.76	0.57	0.75	0.69	0.72	0.54	0.65	0.73
Money Laundering	72	0.64	0.40	0.81	0.49	0.54	0.55	0.63	0.51
Sexual Violence	68	0.73	0.44	0.86	0.20	0.57	0.48	0.65	0.50
Fraud	29	1.00	0.28	0.57	0.24	0.45	0.28	0.75	0.35

TABLE VI
ACCURACY OF THE APPLICATION OF THE BEST FOUR ALGORITHMS ON THE EMBEDDINGS OF M3 ON "MODENATODAY" NEWS ARTICLES.

Algorithm	*Simple average (no lemma)*	*Simple average (lemma)*	*TF-IDF average (no lemma)*	*TF-IDF average (lemma)*
SGD (L2 norm, Hinge loss)	0.84	0.85	0.80	0.82
SVC (RBF kernel, C=1.0, gamma='scale')	0.84	0.84	0.83	0.83
Linear SVC (C=1.0)	0.84	0.85	0.81	0.83
XGBboost	0.83	0.82	0.82	0.82

M3 and the best four categorization algorithms of the previous examples on the news articles published in "ModenaToday".

There are the following two reasons for choosing M3:

- the training of a Word2Vec model from scratch on our dataset requires 15 minutes, while the use of transfer training learning requires less than 3 minutes for retraining,
- the pre-trained model has a wider vocabulary. It could be useful the feature extraction for new news articles which contain words that do not appear in the training corpus. However, it is highly likely that all those words that are discriminative for crime categories are already present in the vocabulary of M2.

Table VI shows the value of accuracy achieved by the best categorization algorithms. Compared to the values of Table IV, we can notice that accuracy is slightly higher if we consider only "ModenaToday" news articles.

B. Unsupervised Text Categorization

The unsupervised text categorization is also known as clustering. This is the task of grouping a set of objects in such a way that objects in the same group (called cluster) are more similar to each other than those in the other groups. The use of clustering for crime categorization consists of feeding the obtained features into an algorithm and checking if the final clusters have a correspondence with the crime categories listed in Section II.

Clustering test is performed on the features obtained by M3, according to the results of the supervised categorization. We decided to use only the news articles published in the "ModenaToday" newspaper since the annotation provided by this newspaper is more reliable than the categorization provided by "Gazzetta di Modena". The dataset contains 5,693 news articles and is unbalanced.

TABLE VII
THE NUMBER OF NEWS ARTICLES FROM "MODENATODAY" NEWSPAPER FOR EACH CATEGORY.

Category	*num. of news articles*
Theft	2314
Drug Dealing	794
Illegal sale	675
Robbery	599
Aggression	416
Scam	400
Murder	177
Kidnapping	160
Mistreatment	85
Evasion	35
Sexual Violence	18
Money Laundering	17
Fraud	3
Total	5693

To address the unbalancing problem, we use the *Synthetic Minority Oversampling Technique* (SMOTE) [26]. The approach is to oversample the elements in the minority class. Starting from an unbalanced dataset, this technique creates new samples for the classes that are present in minority in order to equal the number of elements in the most present category. The algorithm works in the feature space, then the new points do not correspond to real data. SMOTE first selects a minority class instance a at random and finds its k nearest minority class neighbors. The synthetic instance is then created by choosing one of the k nearest neighbors b at random and

TABLE VIII
RESULTS OF UNSUPERVISED TEXT CATEGORIZATION WITH THE APPLICATION OF SPECTRAL CLUSTERING (*n=13*) ON SIMPLE AVERAGE WITHOUT LEMMATIZATION OBTAINED BY M3.

Category	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*
Kidnapping	29	14	1	0	0	9	5	**117**	0	1	10	10	4
Murder	0	15	1	1	0	2	5	0	1	4	**171**	0	0
Robbery	0	2	17	6	**121**	9	8	0	0	25	1	0	11
Theft	2	18	15	8	76	**50**	0	6	3	13	1	0	8
Aggression	0	14	7	3	11	2	58	0	0	**92**	7	0	6
Sexual violence	0	0	**136**	40	0	0	0	0	0	12	5	0	7
Mistreatment	0	9	2	12	0	0	**161**	4	0	3	4	3	2
Scam	23	26	8	2	1	27	3	1	0	1	1	1	**106**
Fraud	56	0	0	0	0	62	0	0	0	0	0	**82**	0
Money laundering	**102**	56	0	0	0	31	0	0	0	1	0	0	10
Illegal sale	3	7	2	0	0	0	0	0	**181**	6	0	0	1
Drug dealing	10	**19**	0	6	1	14	2	5	127	9	3	2	2
Evasion	3	25	0	**137**	0	21	0	0	0	3	5	6	0

connecting a and b to form a line segment in the feature space. The synthetic instances are generated as a convex combination of the two chosen instances a and b.

Table VII shows the number of news articles for each category in the dataset. The most present category is "theft" with 2,314 articles. The least present category is "Fraud" with 3 articles. The algorithm generates 2311 new points for the last category in order to achieve the number of instances in "theft". In the end, in our test, there are 30,082 points in the feature space (2,314 for each category). The algorithm takes too long to cluster these points (more than 30 minutes). So, only 200 instances for each category are involved in the clustering (2,600 total instances). During the selection of the points, priority is given to points corresponding to real newspaper articles. This means that, for the categories which already have more than 200 points before the SMOTE (the first six in Table VII), all the considered points correspond to real newspaper articles. For the other categories, all the real points are considered together with some of the points generated by SMOTE to achieve 200 points for each category.

Four unsupervised algorithms are chosen for our experiments:

- K-means
- Mini Batch K-means
- Agglomerative Clustering
- Spectral Clustering

For all these algorithms, the number of clusters n has to be established in advance. We start by setting *n=13*, that is the number of categories used by the newspaper. We would expect each category to be more present within only one cluster. The best result is given by the Spectral Clustering by using the features generated with the simple mean of the word embeddings without lemmatization. Table VIII shows the result of this test. The rows of the table represent the category, while the columns are the clusters. The elements of the table indicate how many points of each category are inserted in each cluster.

Considering the table column by column, we can notice that all the clusters have some dominant categories. Three clusters have two dominant categories: "Mistreatment" and "Aggression" in the 7th cluster, "Illegal sale" and "Drug dealing" in the 9th cluster, "Theft" and "Robbery" in the 5th cluster. While in the other clusters there is only one dominant category (for example, in the 12th cluster the most present category is "Fraud", while "Scam" is the most present one in the 13th cluster). Considering the table row by row, there are three categories that prevail in more than a cluster: "Fraud", "Theft" and "Money laundering".

To calculate the values of accuracy, precision and recall we need to assign a category to each cluster. We start with the highest number of points for a certain category in a cluster (in our case, the category "Illegal sale" in the 9th cluster). In this way, the category has been assigned to a cluster. Then, we go on with the other clusters and the other categories again starting from the highest number of points. The process assigns only one category to each cluster and a category cannot be assigned to multiple clusters. For each cluster, we calculate the value of accuracy and we find an averaged value of 0.93.

TABLE IX
RESULTS OF UNSUPERVISED TEXT CATEGORIZATION WITH THE APPLICATION OF AGGLOMERATIVE CLUSTERING (*n=7*) ON SIMPLE MEAN WITH LEMMATIZATION OBTAINED BY M3.

Macro-category	*1*	*2*	*3*	*4*	*5*	*6*	*7*
Kidnapping	**158**	23	9	2	5	0	3
Murder	1	18	**166**	0	14	1	0
Robbery, Theft	2	28	1	**268**	74	22	5
Mistreatment, Aggression, Sexual Violence	1	43	23	6	**474**	53	0
Scam, Fraud, Money Laundering	216	**348**	1	31	2	2	0
Illegal sale, Drug dealing	4	23	2	1	6	5	**359**
Evasion	19	19	3	14	3	**142**	0

Analyzing in detail the results of this experiment, we notice that the clusters group together categories that are semantically similar. Based on this consideration, we decided to run a test by grouping together semantically similar categories in macro-category. The chosen macro-categories are seven:

- "Kidnapping",
- "Murder",

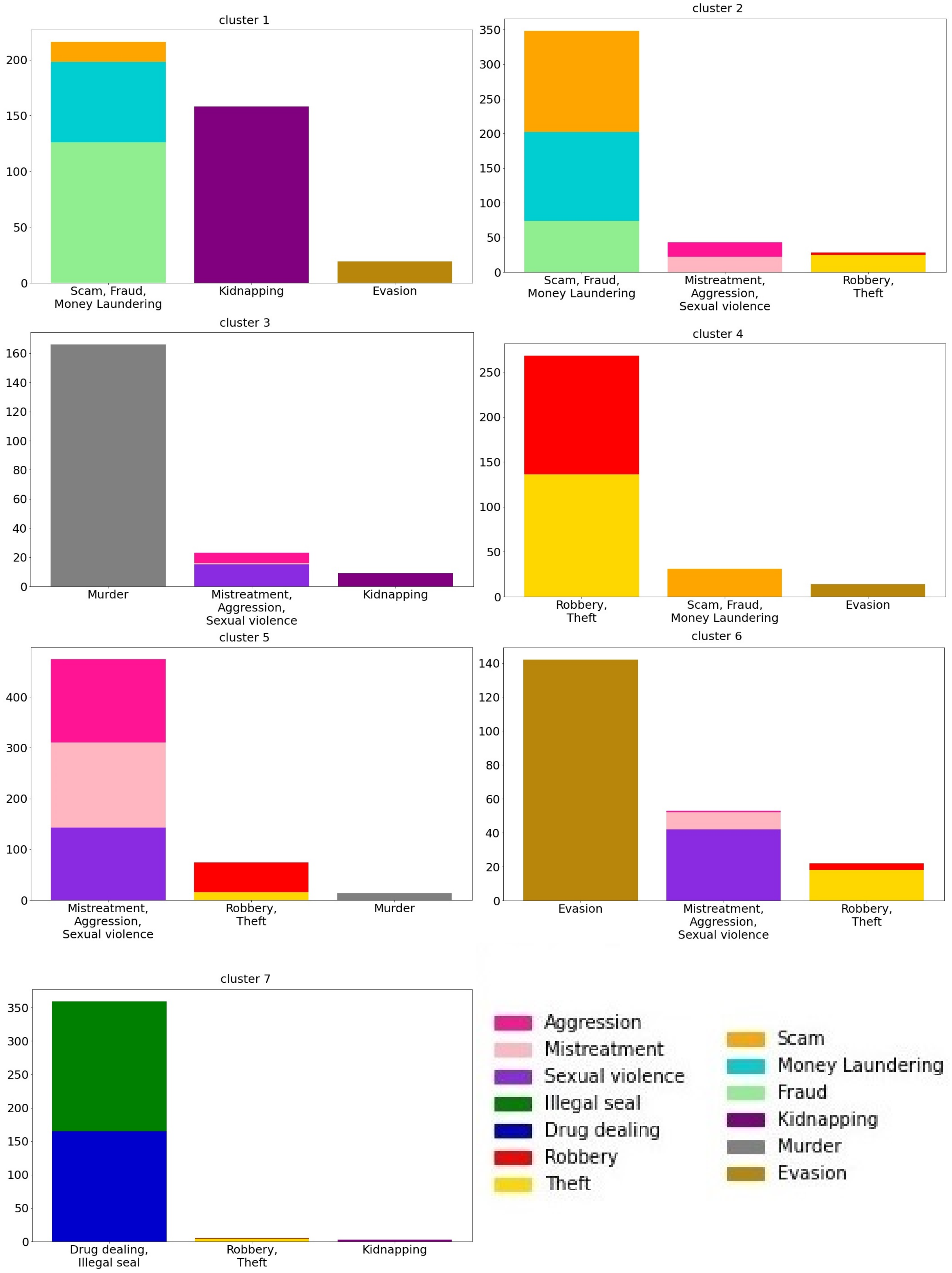


Fig. 2. Histograms with the distribution of crime news articles in the seven clusters obtained by applying the Agglomerative Clustering (*n*=*7*) on the simple mean of the word embeddings with lemmatization obtained by M3.

- “Robbery”, “Theft”,
- “Mistreatment”, “Aggression”, “Sexual Violence”,
- “Scam”, “Fraud”, “Money Laundering”,
- “Illegal Sale”, “Drug Dealing”,
- “Evasion”.

All the four models tested before are re-used to perform categorization with macro-categories and the best result is given by the Agglomerative Clustering using the features generated by the simple average with lemmatization. The results are shown in Table IX. In this case, we get a better result since six out of seven clusters actually have only one dominant category. Furthermore, the macro-category “Scam, Fraud, Money Laundering” is dominant in two different clusters, the first and second ones. The accuracy achieved in this experiment is 0.92. Figure 2 displays the category of news articles contained in each cluster.

VI. Conclusion

In this paper, the use of word embeddings for the crime categorization on an Italian dataset of 15,000 news articles has been proved. Both supervised and unsupervised categorization algorithms have been explored. The model used to obtain word embeddings is Word2Vec, while the categorization algorithms which show the best results are the Linear SVC (supervised text categorization), the Spectral Clustering and the Agglomerative Clustering (unsupervised text categorization). The method described in the paper can be applied also in other contexts and is suitable for documents in languages different from Italian. However, since Word2Vec is language-dependent, it is necessary to use the appropriate Word2Vec model (if exists) or train the model on the documents in the specific language. It also possible to test this approach on word embeddings generated by using other models, such as Glove or FastText. After generating word embeddings, supervised and unsupervised algorithms can be applied as described in the paper.

The results of our experiments show that the representations of texts through word embeddings are suitable for text categorization. Indeed, in all cases, we achieved high accuracy values, greater than 0.80. The results of supervised and unsupervised algorithms have been compared on a subset of 5,683 news articles and show that the supervised approach reaches an accuracy between 0.80 and 0.85, while the unsupervised approach outperforms an accuracy of 0.93. The dataset is available online for further experiments and contains the url of the news articles along with the category provided by the newspapers and the categories assigned by the supervised and unsupervised text categorization.[4]

Both supervised and unsupervised approaches are affected by the imbalance of the dataset and the uncertainty of the annotation provided by the newspapers. In addition, in some cases, news articles are related to general information about crimes and they do not describe a specific crime event. For the first problem, the use of SMOTE technique allows enhancing the results in the unsupervised approach. To overcome the difficulties due to the inaccurate annotation of the newspapers, a manual re-annotation is needed. Since this is a very time-consuming operation, the supervised text categorization can be exploited with the active learning technique. This approach allows categorizing more news articles in a short time without the need for manual checking the annotations predicted by the algorithm with high confidence. This approach will be explored in future work.

[4] https://github.com/SemanticFun/Crime-Text-Categorization

References

[1] S. Ghankutkar, N. Sarkar, P. Gajbhiye, S. Yadav, D. Kalbande, and N. Bakereywala, “Modelling machine learning for analysing crime news,” in *2019 International Conference on Advances in Computing, Communication and Control (ICAC3)*, 2019, pp. 1–5. [Online]. Available: https://doi.org/10.1109/ICAC347590.2019.9036769

[2] M. Hassan and M. Z. Rahman, “Crime news analysis: Location and story detection,” in *2017 20th International Conference of Computer and Information Technology (ICCIT)*, 2017, pp. 1–6. [Online]. Available: https://doi.org/10.1109/ICCITECHN.2017.8281798

[3] D. Velásquez, S. Medina, G. Yamada, P. Lavado, M. Núñez, H. Alatrista, and J. Morzan, “I read the news today, oh boy: The effect of crime news coverage on crime perception and trust,” Institute of Labor Economics (IZA), IZA Discussion Papers 12056, Dec. 2018. [Online]. Available: https://ideas.repec.org/p/iza/izadps/dp12056.html

[4] D. Ghosh, S. A. Chun, B. Shafiq, and N. R. Adam, “Big data-based smart city platform: Real-time crime analysis,” in *Proceedings of the 17th International Digital Government Research Conference on Digital Government Research, DG.O 2016, Shanghai, China, June 08 - 10, 2016*, Y. Kim and S. M. Liu, Eds. ACM, 2016, pp. 58–66. [Online]. Available: https://doi.org/10.1145/2912160.2912205

[5] S. K and P. S. Thilagam, “Crime base: Towards building a knowledge base for crime entities and their relationships from online news papers,” *Information Processing & Management*, vol. 56, no. 6, p. 102059, 2019. [Online]. Available: https://doi.org/10.1016/j.ipm.2019.102059

[6] L. Po and F. Rollo, “Building an urban theft map by analyzing newspaper crime reports,” in *2018 13th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP)*, 2018, pp. 13–18. [Online]. Available: https://doi.org/10.1109/SMAP.2018.8501866

[7] T. Dasgupta, A. Naskar, R. Saha, and L. Dey, “Crimeprofiler: Crime information extraction and visualization from news media,” in *Proceedings of the International Conference on Web Intelligence*, ser. WI ’17. New York, NY, USA: Association for Computing Machinery, 2017, p. 541–549. [Online]. Available: https://doi.org/10.1145/3106426.3106476

[8] F. Rollo and L. Po, "Crime event localization and deduplication," in *The Semantic Web – ISWC 2020*, J. Z. Pan, V. Tamma, C. d'Amato, K. Janowicz, B. Fu, A. Polleres, O. Seneviratne, and L. Kagal, Eds. Cham: Springer International Publishing, 2020, pp. 361–377. [Online]. Available: https://doi.org/10.1007/978-3-030-62466-8_23

[9] L. Po, F. Rollo, and R. T. Lado, "Topic detection in multichannel italian newspapers," in *Semantic Keyword-Based Search on Structured Data Sources - COST Action IC1302 Second International KEYSTONE Conference, IKC 2016, Cluj-Napoca, Romania, September 8-9, 2016, Revised Selected Papers*, ser. Lecture Notes in Computer Science, A. Calì, D. Gorgan, and M. Ugarte, Eds., vol. 10151, 2016, pp. 62–75. [Online]. Available: https://doi.org/10.1007/978-3-319-53640-8_6

[10] F. Rollo, "A key-entity graph for clustering multichannel news: student research abstract," in *Proceedings of the Symposium on Applied Computing, SAC 2017, Marrakech, Morocco, April 3-7, 2017*, A. Seffah, B. Penzenstadler, C. Alves, and X. Peng, Eds. ACM, 2017, pp. 699–700. [Online]. Available: https://doi.org/10.1145/3019612.3019930

[11] S. Bergamaschi, L. Po, and S. Sorrentino, "Comparing topic models for a movie recommendation system," in *WEBIST 2014 - Proceedings of the 10th International Conference on Web Information Systems and Technologies, Volume 2, Barcelona, Spain, 3-5 April, 2014*, V. Monfort and K. Krempels, Eds. SciTePress, 2014, pp. 172–183. [Online]. Available: https://doi.org/10.5220/0004835601720183

[12] L. Po and D. Malvezzi, "Community detection applied on big linked data," *J. Univers. Comput. Sci.*, vol. 24, no. 11, pp. 1627–1650, 2018. [Online]. Available: http://www.jucs.org/jucs_24_11/community_detection_applied_on

[13] C. Wang, P. Nulty, and D. Lillis, "A comparative study on word embeddings in deep learning for text classification," in *Proceedings of the 4th International Conference on Natural Language Processing and Information Retrieval*, ser. NLPIR 2020. New York, NY, USA: Association for Computing Machinery, 2020, p. 37–46. [Online]. Available: https://doi.org/10.1145/3443279.3443304

[14] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2013. [Online]. Available: http://arxiv.org/abs/1301.3781

[15] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, 07 2016. [Online]. Available: https://doi.org/10.1162/tacl_a_00051

[16] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, A. Moschitti, B. Pang, and W. Daelemans, Eds. ACL, 2014, pp. 1532–1543. [Online]. Available: https://doi.org/10.3115/v1/d14-1162

[17] A. Moreo, A. Esuli, and F. Sebastiani, "Word-class embeddings for multiclass text classification," *Data Min. Knowl. Discov.*, vol. 35, no. 3, pp. 911–963, 2021. [Online]. Available: https://doi.org/10.1007/s10618-020-00735-3

[18] A. Fesseha, S. Xiong, E. D. Emiru, M. Diallo, and A. Dahou, "Text classification based on convolutional neural networks and word embedding for low-resource languages: Tigrinya," *Inf.*, vol. 12, no. 2, p. 52, 2021. [Online]. Available: https://doi.org/10.3390/info12020052

[19] A. Borg, M. Boldt, O. Rosander, and J. Ahlstrand, "E-mail classification with machine learning and word embeddings for improved customer support," *Neural Comput. Appl.*, vol. 33, no. 6, pp. 1881–1902, 2021. [Online]. Available: https://doi.org/10.1007/s00521-020-05058-4

[20] E. Christodoulou, A. Gregoriades, M. Pampaka, and H. Herodotou, "Application of classification and word embedding techniques to evaluate tourists' hotel-revisit intention," in *Proceedings of the 23rd International Conference on Enterprise Information Systems, ICEIS 2021, Online Streaming, April 26-28, 2021, Volume 1*, J. Filipe, M. Smialek, A. Brodsky, and S. Hammoudi, Eds. SCITEPRESS, 2021, pp. 216–223. [Online]. Available: https://doi.org/10.5220/0010453502160223

[21] P. Semberecki and H. Maciejewski, "Deep learning methods for subject text classification of articles," in *Proceedings of the 2017 Federated Conference on Computer Science and Information Systems, FedCSIS 2017, Prague, Czech Republic, September 3-6, 2017*, ser. Annals of Computer Science and Information Systems, M. Ganzha, L. A. Maciaszek, and M. Paprzycki, Eds., vol. 11, 2017, pp. 357–360. [Online]. Available: https://doi.org/10.15439/2017F414

[22] T. Lin, "Performance of different word embeddings on text classification," https://towardsdatascience.com/nlp-performance-of-different-word-embeddings-on-text-classification-de648c6262b, 2019, accessed: 7 June 2021.

[23] J. Lilleberg, Y. Zhu, and Y. Zhang, "Support vector machines and word2vec for text classification with semantic features," in *14th IEEE International Conference on Cognitive Informatics & Cognitive Computing, ICCI*CC 2015, Beijing, China, July 6-8, 2015*, N. Ge, J. Lu, Y. Wang, N. Howard, P. Chen, X. Tao, B. Zhang, and L. A. Zadeh, Eds. IEEE Computer Society, 2015, pp. 136–140. [Online]. Available: https://doi.org/10.1109/ICCI-CC.2015.7259377

[24] G. Di Gennaro, A. Buonanno, A. Di Girolamo, A. Ospedale, F. A. N. Palmieri, and G. Fedele, *An Analysis of Word2Vec for the Italian Language*. Singapore: Springer Singapore, 2021, pp. 137–146. [Online]. Available: https://doi.org/10.1007/978-981-15-5093-5_13

[25] B. Li, A. Drozd, Y. Guo, T. Liu, S. Matsuoka, and X. Du, "Scaling word2vec on big corpus," *Data Sci. Eng.*, vol. 4, no. 2, pp. 157–175, 2019. [Online]. Available: https://doi.org/10.1007/s41019-019-0096-6

[26] K. W. Bowyer, N. V. Chawla, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *CoRR*, vol. abs/1106.1813, 2011. [Online]. Available: https://doi.org/10.1613/jair.953

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 471–474

DOI: 10.15439/2021F138
ISSN 2300-5963 ACSIS, Vol. 25

Business Process Recomposition as a Way to Redesign Workflows Effectively

Piotr Wiśniewski, Krzysztof Kluza, Paweł Jemioło, Antoni Ligęza
AGH University of Science and Technology
al. A. Mickiewicza 30, 30-059 Krakow, Poland
E-mail: {wpiotr,kluza,pawljmlo,ligeza}@agh.edu.pl

Anna Suchenia
Cracow University of Technology
ul. Warszawska 24, 31-155 Kraków, Poland
Email: asuchenia@pk.edu.pl

***Abstract*—Business process models are subject to changing requirements. The purpose of this paper is to present methods that enable computer-aided recomposition of process models, understood as using existing processes to design new ones. This procedure involves dividing existing BPMN diagrams into smaller components, from which new models can be created. This kind of model generation can be performed manually by the user or run automatically, based on the Constraint Programming technique. The presented algorithms can improve the process of model redesign and allow users to avoid typical anomalies that may occur in the modeling phase.**

***Index Terms*—Business Process Management, process models, BPMN, process decomposition, process composition**

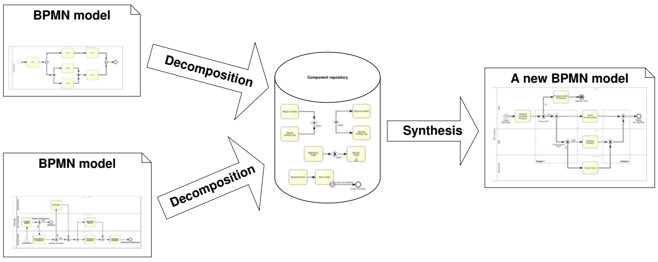


Figure 1. Schema of the idea of recomposing business process models

I. Introduction

BUSINESS process management is a set of methods aimed at designing, analyzing, implementing, and improving task sequences performed in a specific organization. Process models are a method of knowledge representation that helps interested entities visualize and optimize implemented processes, allowing them to achieve their business goals more effectively. There exist many tools supporting knowledge management processes [1]. However, observing process industrial applications [2], many manually created models have quality defects. For this reason, computer-assisted modeling is a valuable technique to eliminate basic errors, the removal of which at the implementation stage is costlier and time-consuming. With this in mind, the work aims to present selected methods allowing to automatically generate a process model based on a diagram repository created due to the division of other models. The concept of process recomposition presented in this work is defined as a combination of decomposition of models into subprojects [3] and synthesis of process models (based on declarative specifications) [4]. The scheme of the described approach is presented in Figure 1.

The concept of decomposition of business process models is mainly used to redesign existing models or create new ones using ready-made components. One of the existing approaches is based on the identification of the largest repeating fragments [5]. The process of recomposition, however, uses formal representations of models that allow their faster and easier reconstruction. Models based on the declarative specification are equally common [6], which instead of describing the sequence explicitly, is based on the constraints and dependencies between the activities in the process.

The useful tools in redesigning the business process are also model repositories that can be presented in the form of a database storing relevant process fragments related to metadata, such as types of objects and possible connections to other fragments [7]. In the case of the synthesis of a model composed of such fragments, it is necessary to ensure the correctness of the generated solution. For example, during automated process modeling, verification of typical anomalies such as deadlock, or loops should be considered [8].

II. Background

A. Business Process Modeling

One of the essential areas of using process models is highly developed IT systems employed in industry and business. One of the most frequently applied notations used to model business processes is BPMN (Business Process Model and Notation). Creating models of functional and technical integration of a part of the system and visualization of the processes supported by the system allows making the specification understandable for all people participating in its implementation.

B. Simple Formal Model of a Business Process

The process model decomposition method used in the presented approach is based on the representation of the BPMN diagram in the form of a directed graph. This type of data structure is considered to best suit the process model [9]. The applied method allows the existence of a loop as well as many start and end events. It can therefore serve as accurate mapping of the model created in the BPMN notation. The graph of the business process is defined by a coherent graph directed $G_P = (V_O, E_F)$, where: V_O is a non-empty set of vertices representing all flow objects, and E_F is a non-empty set of edges representing all connecting objects.

Transforming the BPMN diagram into a graph representation consists of formulating a list of all flow objects in the process along with additional information such as ID and facility name, pool/lane, object type (task, logical gate, etc.), condition (optional) and existing data units (optional). Then, all vertices of the graph should be connected with the addressed edges, to which, in the case of conditional flows (outputs from the XOR or OR gateway), a condition corresponding to a given branch in the process should be attached. A neat form of recording the business process graph is the neighborhood matrix D. Assuming that the considered process consists of n flow objects, D is a $n \times n$ square matrix in which the row and column indexes correspond to the numbers of individual objects. The individual elements of the neighborhood matrix assume a value of 1 when the two vertices are combined and zero in the other cases. Each process graph of matrix D should be accompanied by a detailed specification of flow objects described above.

III. Business Process Recomposition

As it was shown in Figure 1, the recomposition of business processes occurs in three steps: decomposing models, storing model components in a repository, and constructing new diagrams. This section describes in detail each of the phases.

A. Decomposition of process models into sub-diagrams

The business process model decomposition algorithm consists of finding all subgraphs induced by k vertices of the process graph, where $2 \leq k < n$. In the first stage, appropriate k-numerous sets of vertices defining the generated subgraphs should be determined. Then, for each subgraph, a neighborhood matrix is created by removing from the matrix D all rows and columns that do not correspond to the vertices of the selected subset. The simplest example of decomposition is the division of the process graph into all subgraphs with two vertices. From the input graph, each edge with its endpoints is extracted. A more complex problem is the generation of subgraphs consisting of three objects. This type of trigram concerning the vertex v can be defined in one of three ways:

- v with two predecessors,
- v and one predecessor and one successor,
- v with two successors.

To illustrate our method, two BPMN diagrams describing banking processes (Figures 2 and 3) were decomposed.

The first one presents in simplified the procedure for opening a savings account or a savings and settlement account, and the second describes the process of granting a loan or opening a deposit. As part of one process, it is possible to execute its main activities several times, including selecting the right service and launching it. For both models, a decomposition was performed on three-element subgraphs.

The problem of decomposition of process models can also be generalized to the generation of k-element subgraphs induced by each k vertices of the process graph. Its solution is possible by using the Constraint Programming technique with the following input data:

- decision variable: k-element vector v_{Ok} corresponding to a subset of vertices,
- constraints: all elements of the vector are different pairs and connected by edges,

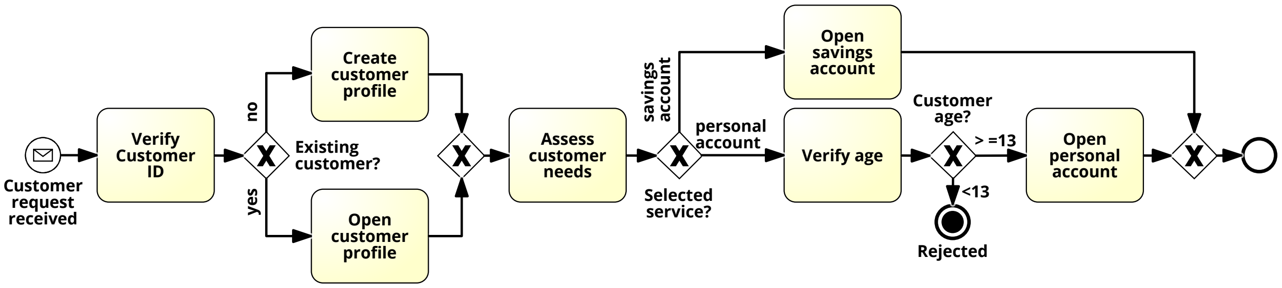


Figure 2. Business process model of opening a bank account

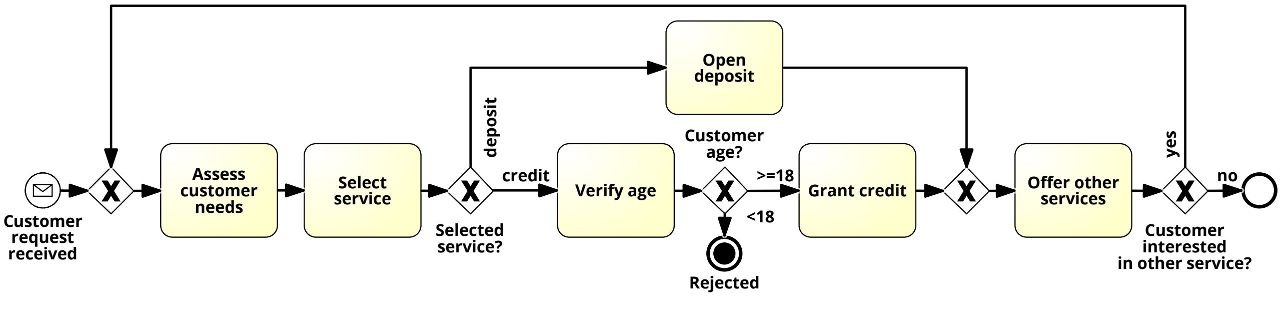


Figure 3. Model of the business process of opening a deposit and granting credit

- solutions: the adjacency matrix of a specific subgraph.

The subgraphs generated that way can be re-converted into BPMN diagrams by performing a reverse algorithm for converting the process model into a directed graph. In other words, each vertex of the subgraph may be changed to the appropriate flow object, while non-zero values of the neighborhood matrix correspond to sequential flows in the process.

Based on the defined groups, it is possible to estimate the number of potential inputs and outputs in each of the generated subgraphs by calculating the existing input and output flows of a given flow object and comparing them with the group to which the given object belongs. This number is used to determine the function $\sigma_{(I)}$, which takes 0 if there are no more entries, 1, if there is one entry, 2, in other cases.

An example of a sub-diagram generated using decomposition of the business process of opening a deposit and granting credit (Figure 3) is shown in Figure 5. It is a component of the MESE type, having many potential entries and exits.

B. The concept of a component repository

The previously described decomposition algorithm generates all possible k-element subgraphs based on the input process graph. This solution means that every possible configuration of k connected flow objects is ready for reuse in new business process models. The disadvantage of this approach is the risk of a situation in which some of the generated solutions may result in the creation of inconsistent models as a result of recomposition due to the imprecision of individual sub-diagrams. In order to ensure the correctness of the models contained in the repository and, at the same time, minimize the risk of omitting significant sub-diagrams, a method of validating the generated components based on the user's assessment was proposed. After the decomposition, a list of potentially inconsistent components is generated. For additional validation, diagrams that meet at least one of the following conditions are selected:

- split gateway without any output branch,
- merge gateway without any input branch,
- number of swimlanes greater than two.

The next step is to classify the subgraphs due to the potential number of entries and exits, as well as the similarity between them. The first of the classifiers consists of grouping models based on the values assumed by the functions $\sigma_{(I)}$ and $\sigma_{(O)}$. The second one is based on the vector technique [10], in which similarity between the two sub-systems depends on the number of common flow objects. Bearing in mind the two presented criteria, selecting the appropriate sub-diagram consists of selecting the appropriate group or specific flow objects, which should be included in the selected diagram. The format of recording sub-patterns in the repository should be conditioned by the method used to synthesize the models later. In the case of manual connection of components, it is recommended to store them in the BPMN 2.0 XML standard, which consists of saving the process model in a file containing definitions of process elements and connections between them. This representation allows you to directly load the file into the graphic editor and facilitate its subsequent modification. The automatic synthesis of processes described in this work uses a graphical representation to generate a complete model. The more practical solution is to leave the sub-diagrams in the form of graphs represented by the neighborhood matrix and the specification of the contained flow objects.

C. Composition of a process model based on the existing components

The last stage of model recompilation occurs by synthesizing BPMN diagrams based on the created subprogram subdiagrams. The first of the proposed methods is the manual creation of a process model in a graphical editor, using a ready-made component database associated with the basic modeling guidelines. In addition to storing generated subgraphs, the purpose of having such a knowledge base is to avoid typical anomalies occurring in process models [8] and reduce the need for formal verification of the final model [11]. Among modeling errors, two basic groups are distinguished:

- syntactic anomalies, such as wrong use of flow objects or swimlanes,
- structural anomalies that concern incorrect behavior of the process during execution, such as deadlocks or infinite loops.

The user interface of the graphic editor should suggest the recommended connections and protect against creating connections that may lead to incorrect execution of the process. An example of an incorrect connection is shown in Figure 6.

An illustration of the process recomposition result using the manual synthesis method is presented in Figure 4. The main assumption was to include in the process of opening a bank account (Figure 2) the possibility of opening a deposit and allowing the situation where the client orders several services within one visit. The colors are marked components extracted as a result of the decomposition of different models.

The synthesis of sub-model process models can also be carried out in an automated way, using a Constraint Programming technique. The modeling person then determines the list of tasks that should be performed as part of the process, and the task of the synthesis algorithm is to create from the available fragments a correct BPMN diagram containing these tasks and meeting the limitations resulting from the adopted notation. The necessary constraints include:

- exactly one start event,
- at least one end event,
- correct number of inputs and outputs of each object,
- closed gateway structures – no possibility of two final events at the same time.

The use of the Constraint Programming method for the modeling problem finds many solutions that meet the given conditions [12]. This approach allows the user to compare several generated models and choose the best result based on the adopted criteria.

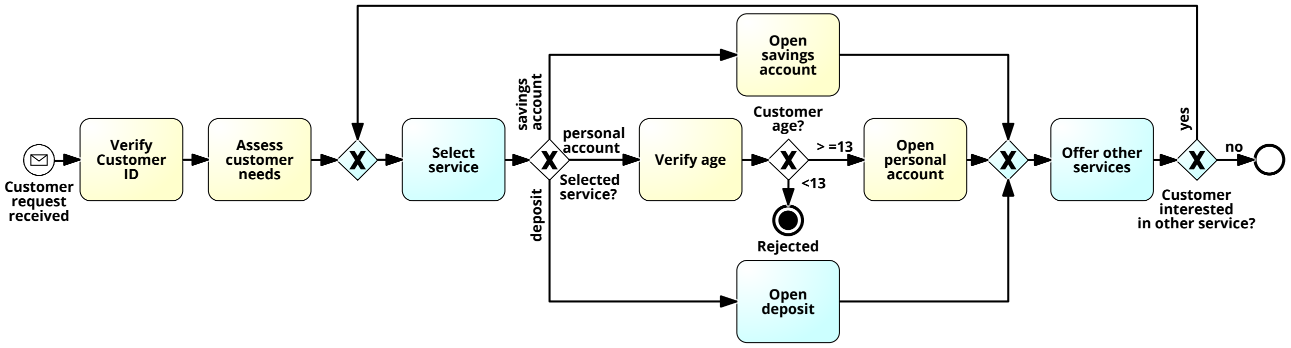


Figure 4. The result of the recomposition of two business processes in the BPMN notation

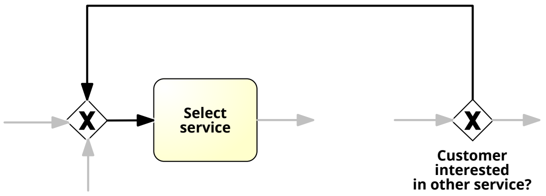


Figure 5. A fragment of a diagram created as a result of decomposition. The gray color indicates potential connections with other process components

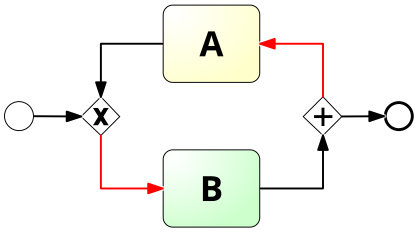


Figure 6. A potentially infinite loop

IV. Conclusions

We presented a method of automatic generation of business process models based on a repository of diagrams created as a result of fragmenting the existing models. Such an approach can be a support for modeling processes by automating the creation of models. Our method consists of synthesizing BPMN diagrams based on the defined components resulting from the decomposition of the processes. It allows a user to create a new, correct model using ready-made fragments of diagrams. An essential element of the recomposition process is the model repository, which allows a user to search for components necessary to synthesize a given diagram efficiently. Implementing a solution supporting automated recomposition of business processes may reduce the amount of time and resources used so far by the organization during process modeling.

In the future, we plan to extend the existing method of component classification in the repository and develop the possibility of synthesizing the process based on a description in natural language. In the case of automatic synthesis of models from sub-diagrams, a significant extension can support the user's decisions in selecting the diagram most suited to the actual process.

References

[1] M. Pondel and J. Pondel, "Selected it tools in enterprise knowledge management processes–overview and efficiency study," in *IFIP International Workshop on Artificial Intelligence for Knowledge Management*. Springer, 2017, pp. 12–28.

[2] H. Leopold, J. Mendling, and O. Günther, "Learning from quality issues of BPMN models from industry," *IEEE software*, vol. 33, no. 4, pp. 26–33, 2015.

[3] P. Wiśniewski, "Decomposition of business process models into reusable sub-diagrams," in *ITM Web of Conferences*, vol. 15. EDP Sciences, 2017, p. 01002.

[4] P. Wiśniewski, K. Kluza, M. Ślażyński, and A. Ligęza, "Constraint-based composition of business process models," in *International Conference on Business Process Management*. Springer, 2017, pp. 133–141.

[5] M. Dumas, L. García-Bañuelos, M. La Rosa, and R. Uba, "Fast detection of exact clones in business process model repositories," *Information Systems*, vol. 38, no. 4, pp. 619–633, 2013.

[6] F. M. Maggi, A. J. Mooij, and W. M. van der Aalst, "User-guided discovery of declarative process models," in *2011 IEEE symposium on computational intelligence and data mining (CIDM)*. IEEE, 2011, pp. 192–199.

[7] M. Skouradaki, V. Andrikopoulos, and F. Leymann, "Representative BPMN 2.0 process model generation from recurring structures," in *2016 IEEE International Conference on Web Services (ICWS)*. IEEE, 2016, pp. 468–475.

[8] A. Suchenia, T. Potempa, A. Ligęza, K. Jobczyk, and K. Kluza, "Selected approaches towards taxonomy of business process anomalies," in *Advances in Business ICT: New Ideas from Ongoing Research*. Springer, 2017, pp. 65–85.

[9] I. M. Weber, *Semantic Methods for Execution-level Business Process Modeling: Modeling Support Through Process Verification and Service Composition*. Springer, 2009, vol. 40.

[10] B. Van Dongen, R. Dijkman, and J. Mendling, "Measuring similarity between business process models," in *Seminal Contributions to Information Systems Engineering*. Springer, 2013, pp. 405–419.

[11] R. Klimek and P. Szwed, "Verification of ArchiMate process specifications based on deductive temporal reasoning," in *2013 Federated Conference on Computer Science and Information Systems*. IEEE, 2013, pp. 1109–1116.

[12] R. Chenouard, L. Granvilliers, and R. Soto, "Model-driven constraint programming," in *Proceedings of the 10th international ACM SIGPLAN conference on Principles and practice of declarative programming*, 2008, pp. 236–246.

Software, System and Service Engineering

THE S3E track emphasizes the issues relevant to developing and maintaining software systems that behave reliably, efficiently and effectively. This track investigates both established traditional approaches and modern emerging approaches to large software production and evolution.

For decades, it is still an open question in software industry, how to provide fast and effective software process and software services, and how to come to the software systems, embedded systems, autonomous systems, or cyber-physical systems that will address the open issue of supporting information management process in many, particularly complex organization systems. Even more, it is a hot issue how to provide a synergy between systems in common and software services as mandatory component of each modern organization, particularly in terms of IoT, Big Data, and Industry 4.0 paradigms.

In recent years, we are the witnesses of great movements in the area of software, system and service engineering (S3E). Such movements are both of technological and methodological nature. By this, today we have a huge selection of various technologies, tools, and methods in S3E as a discipline that helps in a support of the whole information life cycle in organization systems. Despite that, one of the hot issues in practice is still how to effectively develop and maintain complex systems from various aspects, particularly when software components are crucial for addressing declared system goals, and their successful operation. It seems that nowadays we have great theoretical potentials for application of new and more effective approaches in S3E. However, it is more likely that real deployment of such approaches in industry practice is far behind their theoretical potentials.

The main goal of Track 5 is to address open questions and real potentials for various applications of modern approaches and technologies in S3E so as to develop and implement effective software services in a support of information management and system engineering. We intend to address interdisciplinary character of a set of theories, methodologies, processes, architectures, and technologies in disciplines such as: Software Engineering Methods, Techniques, and Technologies, Cyber-Physical Systems, Lean and Agile Software Development, Design of Multimedia and Interaction Systems, Model Driven Approaches in System Development, Development of Effective Software Services and Intelligent Systems, as well as applications in various problem domains. We invite researchers from all over the world who will present their contributions, interdisciplinary approaches or case studies related to modern approaches in S3E. We express an interest in gathering scientists and practitioners interested in applying these disciplines in industry sector, as well as public and government sectors, such as healthcare, education, or security services. Experts from all sectors are welcomed.

Topics

Submissions to S3E are expected from, but not limited to the following topics:

- Advanced methodology approaches in S3E – new research and development issues
- Advanced S3E Process Models
- Applications of S3E in various problem domains – problems and lessons learned
- Applications of S3E in Lean Production and Lean Software Development
- Total Quality Management and Standardization for S3E
- Artificial Intelligence and Machine Learning methods in advancing S3E approaches
- S3E for Information and Business Intelligence Systems
- S3E for Embedded, Agent, Intelligent, Autonomous, and Cyber-Physical Systems
- S3E for Design of Multimedia and Interaction Systems
- S3E with User Experience and Interaction Design Methods
- S3E with Big Data and Data Science methods
- S3E with Blockchain and IoT Systems
- S3E for Cloud and Service-Oriented Systems
- S3E for Smart Data, Smart Products, and Smart Services World
- S3E in Digital Transformation
- Cyber-Physical Systems (8th Workshop IWCPS-8)
- Software Engineering (41th IEEE Workshop SEW-41)
- Advances in Programming Languages (8th Workshop WAPL'21)

Track Chairs

- **Luković, Ivan,** Unniversity of Belgrade, Serbia
- **Kardas, ,** Geylani, Ege University International Computer Institute, Turkey
- **Mazzara, Manuel,** Innopolis University, Russia

Program Chairs

- **Bowen, Jonathan,** Museophile Ltd., United Kingdom
- **Hinchey, Mike** (Lead Chair), Lero-the Irish Software Engineering Research Centre, Ireland
- **Szmuc, Tomasz,** AGH University of Science and Technology, Poland
- **Zalewski, Janusz,** Florida Gulf Coast University, United States

- **Seyed Hossein Haeri,** Catholic University of Louvain, Louvain-la-Neuve, Belgium and University of Bergen, Norway

PROGRAM COMMITTEE

- **Ahmad, Muhammad Ovais,** Karlstad University, Sweden
- **Challenger, Moharram,** University of Antwerp, Belgium
- **Dejanović, Igor,** University of Novi Sad, Faculty of Technical Sciences, Serbia
- **Derezinska, Anna,** Warsaw University of Technology, Poland
- **Dutta, Arpita,** NIT ROURKELA, India
- **García-Mireles, Gabriel,** Universidad de Sonora, Mexico
- **Göknil, Arda,** SINTEF Digital, Norway
- **Heil, Sebastian,** Technische Universitüt Chemnitz, Germany
- **Erata, Ferhat,** Yale University, United States
- **Escalona, M.J.,** University of Seville, Spain
- **Essebaa, Imane,** Faculté des Sciences et Techniques Mohammedia, Morocco
- **Hanslo, Ridewaan,** University of Pretoria, South Africa
- **Jarzebowicz, Aleksander,** Gdansk University of Technology, Poland
- **Kaloyanova, Kalinka,** University of Sofia, Bulgaria
- **Karolyi, Masaryk University,** IBA, Czechia
- **Katic, Marija,** University of London, United Kingdom
- **Khlif, Wiem,** FSEGS, Tunisia
- **Kolukısa Tarhan, Ayça,** Hacettepe University, Turkey
- **Krdzavac, Nenad,** University of Belgrade, Serbia
- **Marcinkowski, Bartosz,** University of Gdansk, Poland
- **Milosavljevic, Gordana,** University of Novi Sad, Faculty of Technical Sciences, Serbia
- **Misra, Sanjay,** Covenant University, Nigeria
- **Morales Trujillo, Miguel Ehécatl,** University of Canterbury, New Zealand
- **Ozkan, Necmettin,** Kuveyt Turk Participation Bank, Turkey
- **Ozkaya, Mert,** Yeditepe University, Turkey
- **Ristic, Sonja,** University of Novi Sad, Faculty of Technical Sciences, Serbia
- **Rossi, Bruno,** Masaryk University, Czech Republic
- **Sanden, Bo,** Colorado Technical University, United States
- **Shilov, Nikolay,** Innopolis University, Russia
- **Sierra Rodríguez, José Luis,** Universidad Complutense de Madrid, Spain
- **Torrecilla-Salinas, Carlos,** IWT2, Spain
- **Varanda Pereira, Maria João,** Instituto Politécnico de Bragança, Portugal

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 477–484
DOI: 10.15439/2021F100
ISSN 2300-5963 ACSIS, Vol. 25

An Empirical Study on Application of Word Embedding Techniques for Prediction of Software Defect Severity Level

Lov Kumar
Dept. CSIS
BITS Pilani Hyderabad Campus
lovkumar505@gmail.com

Mukesh Kumar
Dept. CS&E
NIT Patna
mukesh.kumar@nitp.ac.in

Lalita Bhanu Murthy
Dept. CSIS
BITS Pilani Hyderabad Campus
bhanu@hyderabad.bits-pilani.ac.in

Prof. Sanjay Misra
Østfold University College, Halden, Norway
ssopam@gmail.com

Vipul kocher
Testaing.Com
vipulkocher@testAing.com

Srinivas Padmanabhuni
Testaing.Com
srinivas@testaing.com

Abstract—**Software defect severity level helps to indicate the impact of bugs on the execution of the software and how rapidly these bugs need to be addressed by the team. The working team is regularly analyzing the bugs report and prioritizing the defects. The manual prioritization of these defects based on the experience may be an inaccurate prediction of the severity that will delay in fixing of critical bugs. It is compulsory to automate the process of assigning an appropriate level of severity based on bug report results with an objective to fix critical bugs without any delay. This work aims to develop defect severity level prediction models that have the ability to assign severity level of defects based on bugs report. In this work, seven different word embedding techniques are applied to defect description to represent the word, not just as a number but as a vector in n-dimensional space in order to reduce the number of features. Since the predictive ability of the developed models depends on the vectors extracted from text as they are used as an input to the defect severity level prediction models. Further, three feature selection techniques have been applied to find the right set of relevant vectors. The effectiveness of these word embedding techniques and different sets of vectors are evaluated using eleven different classification techniques with Synthetic Minority Oversampling Technique (SMOTE) to overcome the class imbalance problem. The experimental results show that the word embedding, feature selection techniques and SMOTE have the ability to predict the severity level of the defect in a software.**

Keywords*—*Defect Severity Level Prediction, Data Imbalance Methods, Feature Selection, Classification Techniques, Word Embedding.

I. Introduction

APPLYING data mining techniques on software repositories such as software fault prediction, maintainability prediction, version control systems, source code analysis, bug archives, etc. is an emerging field that has received significant research interest in recent times. Researchers have proposed many tools and methods using machine learning techniques to assist a practitioner in decision making and automating software engineering tasks [1][2][3][4]. However, Forrest et al. observed that the finding and fixing defects in software is a time-consuming and expensive process. They have found that the median time to repair bugs for ArgoUML software is 190 days, and PostgreSQL is 200 days. They have also observed that more than 50% of all fixed bugs in Mozilla took more than 29 days [5][6]. Therefore, it becomes essential to reduce the time and cost of the bug fixing process and also improve the quality of the software system. Defect severity level prediction has been emerged as a novel research field for the effective allocation of resources and plans to fix the defects based on their severity level [3]. These models help to find the severity level of defects that can be used to find the effect of defects on the software. Defect severity level prediction models are designed based on the features extracted from the defect description. Recent research has used different data mining techniques to extract numerical features from defect descriptions for the severity level of defect prediction using machine learning techniques. However, there are three main technical challenges in building defect severity level prediction models for predicting the proper severity level of the defects using defect description.

- **Word Embedding:** The defect severity level prediction models are often developed based on the unstructured form of the description of defects. The unstructured nature of data poses intrinsic challenges. If some sort of numerical features can be assigned using text mining techniques that can use as an input for model development, then it can be utilized for prediction of future severity level of defects. In this work, seven different word embedding techniques such as Continuous Bag of Words Model (CBOW)[1], Skip-gram (SKG)1, Global Vectors for Word Representation (GLOVE)[2], Google news word to vector(w2v)[3], fasttext (FST)[4], Bidirectional Encoder Representations from Transformers (BERT) [5], and generative pre-

[1] https://towardsdatascience.com/nlp-101-word2vec-skip-gram-and-cbow-93512ee24314

[2] https://nlp.stanford.edu/projects/glove/

[3] https://code.google.com/archive/p/word2vec/

[4] https://fasttext.cc/

[5] https://en.wikipedia.org/wiki/BERT_(language_model)

training model (GPT) [6] have been applied on bugs reports to represent the word not just as a number but as a vector in n-dimensional space. The above techniques provide similar representation for similar words and also provide a small number of features as compared to the size of the vocabulary. We have also removed stop-words, spaces, and bad symbols before applying these techniques. The predictive ability of these techniques is compared with frequently used term frequency, inverse document frequency (TFIDF).

- **High-Dimensional Features Data:** The predictive ability of defect severity level prediction models also depends on the features that are considered as the input of the models. Researchers have concluded that the data having high dimension features consisting of redundant and irrelevant features negatively affect the performance of the defect severity level prediction models [2][3][1]. The presence of a huge number of the feature in the case of text analysis poses intrinsic challenges to develop models for predicting the proper severity level of the defects using defect description. In this study, we have used three different feature selection techniques to remove irrelevant features and select the right sets of the relevant features.
- **Imbalanced Data:** The last challenge in building defect severity level prediction model is that the data used for building the models are imbalanced. A dataset is defined as a balanced dataset when the samples of the dependent variable or output variable are approximately evenly distributed across different values of dependent variable [7][8]. In this study, the considered datasets are observed to be not possessing have an equal number of the severity level of the defects. Hence, it has been proposed to apply Synthetic Minority Oversampling Technique (SMOTE) on each dataset in order to get balance data.

Prioritization of defect based on the severity level computed using bugs reports? is a problem encountered by software practitioners and the study presented in this work is motivated by the need to develop defect severity level prediction models using extracted features with the help of word embedding techniques from bugs reports. This study aims to find the best word embedding technique by comparing the predictive ability of the models developed using seven different types of word embedding techniques. It further investigates the application of feature selection techniques, data sampling techniques, and eleven different classification techniques for prediction of severity level of defects.

II. Related Work

Software researchers have used different methods in the past to extract features from bugs report and used these features as an input for developing models. Menzies and Marcus have used various text mining concepts to extract features from the bugs report [9]. They proposed an automated method called SEVERIS and validated these models using the defects report of NASA's Project and Issue Tracking System (PITS). These proposed models help to predict proper severity level of the defects using defect description. Rajni Jindal et al. also done similar work to extract features from defect descriptions using Term Frequency and Inverse Document Frequency (TFIDF) to extract features [10]. They have used the Radial Basis function network for developing defect severities prediction models. Finally, they found that the proposed methods have a high predictive ability to predict the severity levels of the defects. Sari and Siahaan have also followed a similar method for developing models to predict the severity level of the defects based on defect description [11]. They have applied InfoGain gain on extracted features from text to find relevant features for model development. Finally, they have used a support vector machine with an objective to develop defect severities prediction models.

In 2011, David Lo and the team analyzed the performance of models at three different levels of severity: low, medium, and high. It was found that an artificial neural network (ANN) was among the best methods. However, the predictions were less accurate for high severity faults. In 2012, Sharma et al. [12] proposed a priority prediction method using SVM, Naive Bayes, KNN, and Neural Network. This predicted the priority of the newly arrived bug reports, and the accuracy of almost all techniques (except NB) was less than 70% for Eclipse and Open Office projects. In 2014, Gayathri and Sudha developed an enhanced Multilayer Perceptron Neural Network [13]. Comparative analysis of modeling of defect proneness predictions using a dataset of different metrics from NASA MDP (Metrics Data Program) was performed. In 2017, Gupta and Saxena developed a model for the prediction of the existence of bugs in class [14]. The model developed was the object-oriented Software Bug Prediction System (SBPS), and it was trained using the Promise Software Engineering Repository. The Logistic Regression Classifier provided the best accuracy. The average accuracy of the model was found out to be 76.27%.

In the context of software severity level prediction, most of the researchers have used count vectorization and TFIDF to extract numerical features from bugs report. The concepts of these techniques are based on bag-of-words, therefore it has not capability to capture the position of vocabulary in sentences. These methods do not play well with many machine learning models because of high-dimensional features. While in this work, we are attempting to use seven different word embedding techniques that represent the word not just as a number but as a vector in n-dimensional space. The above techniques provide similar representation for similar words. The effectiveness of these word embedding techniques are evaluated using eleven different classification techniques with Synthetic Minority Oversampling Technique (SMOTE) to overcome the class imbalance problem.

III. Study Design

This section presents the details regarding various design setting used for this research.

A. Experimental Dataset

In this study, six different software datasets have been used, which are referred to as CDT, JDT, PDE, Platform, Bugzilla, and Thunderbird to validate our proposed models.

[6] https://openai.com/blog/language-unsupervised/

These datasets have been collected from msr2013-bug_dataset-master [7]. Mining Software Repositories (MSR) conducted Challenge every year by providing software-related data and motivate participate to apply data mining techniques for finding important patterns. The datasets are the collection of bugs reports wherein each bugs report contains the defect ID, defect description, and severity level of the defects. Table I shows the details of the dataset used for this study. As shown in Table I, the CDT software bugs report consists of 2220 normal defects, 146 minor defects, 288 major defects, 42 trivial defects, 58 blocker defects, 106 critical defects.

TABLE I: Experimental Data Set Description

	Normal	Minor	Major	Trivial	Blocker	Critical
CDT	2220	146	288	42	58	106
JDT	1906	261	430	104	50	106
PDE	2380	91	295	52	27	70
Platform	1485	215	715	145	77	215
Bugzilla	1342	598	352	302	167	114
Thunderbird	1100	387	655	91	25	658

B. Training of Models from Imbalanced Data Set:

After analyzing experimental data as shown in Table I, it is quite evident that the considered datasets suffering from class imbalance problem, i.e., the number of samples in each class, are not same. Therefore, balancing of data is required before applying any classification techniques [15]. This approach help to improve the predictive ability of the developed software defect severity level prediction models [16][17]. In this study, we have performed Synthetic Minority Oversampling Technique (SMOTE) one each dataset in order to get balance data. SMOTE technique is identified as a very popular technique by different researches that helps to improve the predictive ability of the models.

C. Word Embedding:

The software bugs report consist of the defect ID and their corresponding defect description. In this work, seven different word embedding techniques including Continuous Bag of Words Model (CBOW), Skip-gram(SKG), Global Vectors for Word Representation (GLOVE), Google news word to vector(w2v), fasttext (FST), Bidirectional Encoder Representations from Transformers (BERT), and generative pre-training model (GPT) have been applied on defect description extracted from bugs reports. We have applied these techniques to represent the word not just as a number but as a vector in n-dimensional space. These vectors are used as input to develop models for assigning appropriate severity levels to the defects present in the bugs reports. We have also removed stopwords, bad symbols, and spaces before applying word embedding. We have also compared the predictive ability of these techniques with term frequency, inverse document frequency(TFIDF).

D. Feature Selection Techniques

After successfully finding a vector of defect description, we have used these vectors as an input of the models. Since we are using these vector of n-dimension as an input of the models, so, the performance of the models also depends upon the selection of important features vectors. In this study, we have used three different features selection techniques, i.e., significant sets of features using rank-sum test, uncorrelated sets of features using cross-correlation analysis, and principal component analysis to remove irrelevant features and select right sets of the relevant feature. We have also compared the predictive ability of the models developed using selected sets of features with original features.

E. Classification Technique:

The predictive ability of different word-emending techniques, feature selection techniques and SMOTE are evaluated using eleven most frequently used classifiers such as multinomial naive bayes (MNB), bernoulli naive bayes (BNB),

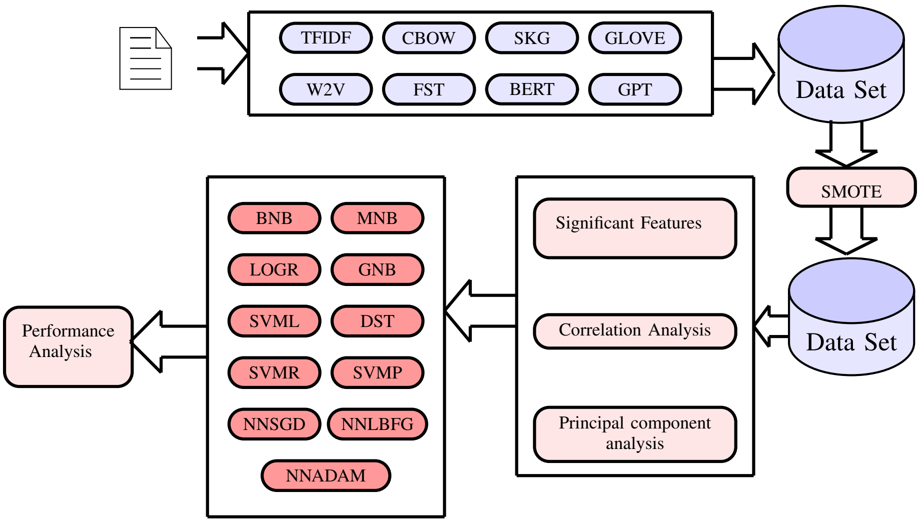


Fig. 1: Framework of proposed work

[7] http://2013.msrconf.org/

gaussian naive bayes (GNB), Logistic Regression (LOGR), decision tree (DST), SVM with linear kernel (SVML), SVM with polynomial kernel (SVMP), SVM with RBF kernel (SVMR), Neural network with LBFG (NNLBFG), Neural network with SGD (NNSGD), and Neural network with ADAM (NNADAM) in software engineering domain [1][18][19].

IV. Research Methodology

In this work, we have applied seven different word embedding methods to extract features from bugs reports and considered these features as an input to develop models for predicting proper severity level of the defects using defect description. These models are trained using eleven different classifiers and validated using 5-fold cross-validation. In this study, we have also considered SMOTE for handling imbalanced data and three feature selection techniques for finding the best combination of relevant features. The detailed overview of our proposed work is giving in Figure 1. The information presented in Figure 1 suggested that the proposed framework is a multi-step process consisting of features extraction from text data using word embedding, handling class imbalance problem using SMOTE, removal of irrelevant features, and finally development of prediction models using eleven different classification techniques.

First, bugs report for a software project is collected from the Bugzilla bug tracking system containing the unique id of defects, description of the defect, and associated severity level of the defects. Next, we have used seven different word embedding to find the numerical representation of defect description. Next, we have used SMOTE techniques to handle the class imbalance problem because the considered dataset is not evenly distributed. The performance of models trained using balanced data is also compared with models developed using original data. After balancing the data, three different features selection techniques such as significant features using rank-sum test, cross-correlation analysis, and principal component analysis are used to remove irrelevant features and select the right sets of reverent features. Finally, eleven different classifiers are used to develop models predicting proper severity level of the defects using defect description. The performance of these developed models is computed and compared using AUC, F-Measure, and accuracy performance values.

V. Empirical Results and Analysis

In this work, we have applied eight different word embedding, one sampling technique, three feature selection techniques, and eleven classification techniques for developing models to predict proper severity level of the defects using defect description. Each word-embedding is applied on the considered datasets as mentioned in Table I. The effectiveness of these word-emending techniques is evaluated using 11 different most frequently used classifiers. Therefore, a total of 4224 (6 datasets * 8 word-embedding*(1 Original Data+ Smote data)*(3 Feature Selection+ 1 All Features)* 11 different classification technique) distinct prediction models are built in the study. The predictive ability of these trained models are evaluated in terms of AUC, F-Measure, and accuracy performance values. These models are validated with the help of 5-fold cross-validation methods. Table II reports the results achieved by different classifiers on original data and sampled data on different sets of features. The results for other cases are of similar type. Looking at information present in Table II, we can be inferred that:

- The high value of AUC confirm that the developed models have the ability to predict proper severity level of the defects using defect description.
- The models developed using a support vector machine with polynomial kernel have better predictive ability as compared to other classifiers.
- The models trained using neural network with ADAM (NNADAM) training algorithm have better predictive ability as compared LBFG, and SGD traing algorithms.
- The models trained by considering balanced data using smote as an input have better predictive ability as compared to original data.

VI. Comparative Analysis

In this section, we analyze and compare the performance of models developed using different word-embedding, classifiers, sampling techniques, and sets of features. In this paper, we have considered Descriptive statistics, box-plot, and Significant tests to compare the developed models for severity level prediction.

A. Word Embedding

The predictive ability of developed defect severity level prediction models using different word embedding are computed with the help of AUC, F-Measure, and accuracy. They are compared using Descriptive statistics, box-plot, and Significant tests. In this study, seven different word embedding techniques such as Continuous Bag of Words Model (CBOW), Skip-gram (SKG), Global Vectors for Word Representation (GLOVE), Google news word to vector(w2v), fasttext (FST), BERT, and generative pre-training model (GPT) have been used to compute the numerical vector of defects reports.
Comparison of Word Embedding: box-plots: Figure 2 provides the performance value, i.e., AUC, F-Measure, and accuracy of different word embedding in terms of Box-Plot diagrams and descriptive statistics. It is clear from Figure 2 that the models developed by considered word vector computed using GLOVE and w2v have better predictive ability to predict the appropriate severity level to the defects present in the bugs reports as compared to other models. The models developed using w2v achieve 0.70 average AUC value, 0.99 max auc, and 0.87 Q3 AUC i.e., 25% models developed using w2v have 0.87 AUC value. However, the models developed using SKG have low predictive ability as compared to other techniques.

Comparison of Word Embedding: Significant Test: In this study, the Wilcoxon signed-rank test is also applied on the AUC, F-Measure, and accuracy for statistically comparing the ability to predict the appropriate severity level of developed models using different word embedding. The objective of this testing is to find whether the models developed using different word embedding have a significant improvement or not. This test uses p-value to accept or reject the considered null hypothesis. The considered null hypothesis for this paper is "the defect severity level prediction models developed by

TABLE II: Performance Value: Classification Techniques

	Accuracy										
	OD										
	MNB	BNB	GNB	LOGR	DST	SVML	SVMP	SVMR	NNLBFG	NNSGD	NNADAM
TFIDF	78.11	75.56	22.78	78.22	62.44	78.22	77.78	78.00	72.67	78.11	73.89
CBOW	78.11	78.11	73.44	78.11	60.22	78.11	70.89	78.11	72.78	78.11	78.00
SKG	78.11	78.00	60.11	78.00	60.22	78.11	74.56	78.11	74.56	78.11	77.67
GLOVE	78.11	78.00	51.00	77.67	59.56	78.00	71.11	78.00	70.22	78.11	73.89
W2V	78.11	77.67	55.11	78.11	60.11	77.89	70.44	78.22	68.56	78.11	75.33
FST	78.11	77.56	24.22	78.00	58.56	78.11	75.11	78.11	72.89	78.11	78.11
BERT	74.22	76.89	17.22	78.11	59.67	78.22	72.11	78.22	78.11	78.11	78.11
GPT	54.67	73.89	7.33	78.11	72.33	78.11	78.00	78.11	78.11	78.11	78.11
	SMOTE										
TFIDF	60.18	62.60	59.50	64.35	79.53	67.47	86.95	88.85	85.78	83.32	91.9
CBOW	42.94	16.18	58.21	54.38	76.17	48.53	91.92	95.92	90.46	92.06	95.71
SKG	24.55	16.21	25.92	44.62	74.39	42.32	75.95	68.59	72.14	58.24	74.71
GLOVE	45.89	16.43	56.77	76.52	78.60	78.29	95.79	98.28	92.09	95.60	95.65
W2V	46.52	16.22	59.43	79.95	77.56	80.43	95.07	98.18	91.88	95.93	94.08
FST	27.45	15.97	34.14	37.29	77.35	33.71	86.66	82.33	67.96	84.94	87.25
BERT	24.13	15.97	31.22	77.99	78.90	81.07	94.72	84.63	15.97	15.97	15.97
GPT	22.49	19.05	27.93	44.52	68.06	46.57	61.16	50.58	15.97	15.97	15.97
	AUC										
	OD										
TFIDF	0.51	0.51	0.59	0.51	0.52	0.51	0.51	0.50	0.53	0.50	0.54
CBOW	0.50	0.50	0.53	0.50	0.55	0.50	0.56	0.50	0.52	0.50	0.50
SKG	0.50	0.50	0.58	0.51	0.57	0.50	0.56	0.50	0.54	0.50	0.51
GLOVE	0.50	0.51	0.64	0.51	0.56	0.50	0.54	0.50	0.53	0.50	0.55
W2V	0.50	0.50	0.63	0.53	0.54	0.51	0.57	0.50	0.57	0.50	0.54
FST	0.50	0.51	0.54	0.51	0.55	0.50	0.52	0.50	0.51	0.50	0.50
BERT	0.53	0.52	0.57	0.51	0.54	0.50	0.53	0.50	0.50	0.50	0.50
GPT	0.57	0.52	0.54	0.50	0.52	0.50	0.50	0.50	0.50	0.50	0.50
	SMOTE										
TFIDF	0.80	0.82	0.80	0.80	0.87	0.82	0.90	0.93	0.92	0.91	0.95
CBOW	0.67	0.51	0.76	0.76	0.85	0.72	0.95	0.98	0.94	0.96	0.98
SKG	0.56	0.50	0.55	0.71	0.84	0.70	0.87	0.83	0.87	0.78	0.88
GLOVE	0.72	0.50	0.78	0.88	0.87	0.89	0.97	0.99	0.96	0.97	0.97
W2V	0.73	0.51	0.79	0.90	0.86	0.91	0.97	0.99	0.96	0.97	0.96
FST	0.57	0.50	0.60	0.64	0.87	0.65	0.93	0.90	0.82	0.92	0.93
BERT	0.56	0.50	0.58	0.88	0.87	0.90	0.97	0.91	0.50	0.50	0.50
GPT	0.55	0.54	0.59	0.70	0.81	0.73	0.78	0.72	0.50	0.50	0.50

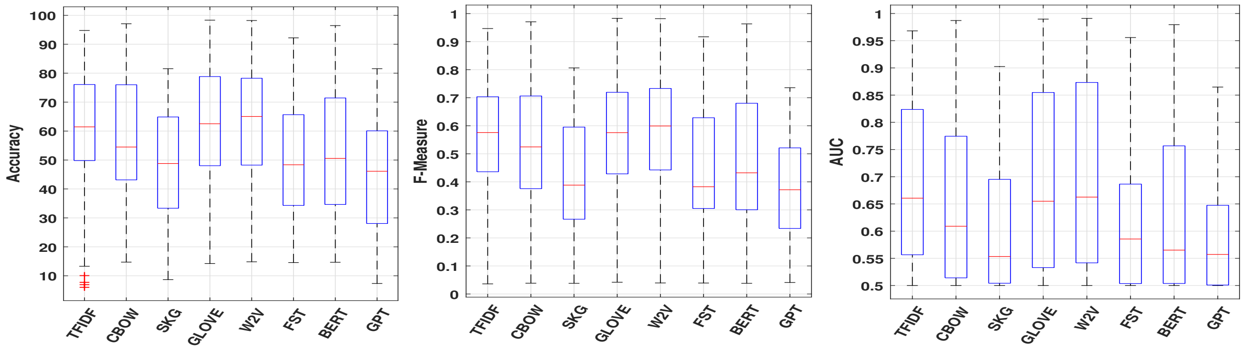


Fig. 2: Performance Box-Plot Diagram: Performance of Different Word Embedding

considering word vector using a different word embedding as an input are significantly same". The considered null hypothesis is only accepted if the obtained p-values using Wilcoxon signed-rank test is greater than 0.05. The results of Wilcoxon signed-rank test on different pairs of word embedding are depicted in Table III. For the purpose of simplicity, we have used only two number for representing results, i.e., 0 means hypothesis accepted (models are significantly same) and 1 means hypothesis rejected (models are significantly different). According to the information present in Table III, the models developed by considering word vector using different word embedding as an input are significantly different for most of the cases.

TABLE III: Significant tests: Different Word Embedding

	TFIDF	CBOW	SKG	GLOVE	W2V	FST	BERT	GPT
TFIDF	0	1	1	0	0	1	1	1
CBOW	1	0	1	1	1	1	1	1
SKG	1	1	0	1	1	0	0	0
GLOVE	0	1	1	0	0	1	1	1
W2V	0	1	1	0	0	1	1	1
FST	1	1	0	1	1	0	0	1
BERT	1	1	0	1	1	0	0	1
GPT	1	1	0	1	1	1	1	0

B. *SMOTE*

The predictive ability of developed defect severity level prediction models using original data and smote sampled data are computed using AUC, F-Measure, and accuracy performance values and compared using Descriptive statistics, box-plot, and Significant tests.
Comparison of Original Data and SMOTE: box-plots: Figure 4 provides the performance value, i.e., AUC, F-Measure, and accuracy of the models developed using original data and smote sampled data in terms of Box-Plot diagrams and descriptive statistics. The information in Figure 4 demonstrate that the SMOTE data sampling technique plays an important role in improving the predictive ability of the defect severity level prediction models. The models developed using SMOTE sampled data achieve 0.75 average AUC value, 0.99 max auc, and 0.86 Q3 AUC, i.e., 25% models developed using SMOTE sampled data have 0.86 AUC value.

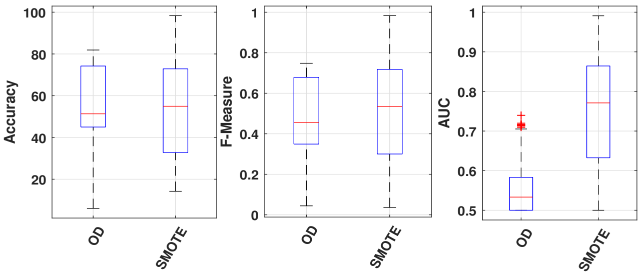


Fig. 4: Performance Box-Plot Diagram: Performance of Original Data and SMOTE

Comparison of Original data SMOTE: Significant Test: In this study, the Wilcoxon signed-rank test is also applied on the AUC, F-Measure, and accuracy for statistically comparing the ability to predict the appropriate severity level of developed models using original data and SMOTE sampled data. The objective of this testing is to find whether the models developed using sampled data have a significant improvement or not. The considered null hypothesis for this paper is "the defect severity level prediction models trained using sampled have not a significant improvement." The considered null hypothesis is only accepted if the obtained p-values using the Wilcoxon signed-rank test is greater than 0.05. In this work, the p-value of the models trained using sampled data and original data is less than 0.05, i.e., our considered hypothesis is rejected. Hence, the models trained using sampled data have significant improvement in predicting defect severity levels.

C. *Feature Selection*

In this study, we have used three different features selection techniques, i.e., significant sets of features using rank-sum test, uncorrelated sets of features using cross-correlation analysis, and principal component analysis to remove irrelevant features and select right sets of the relevant feature. We have also validated the performance of the models developed using selected sets of features with all features using AUC, F-Measure, and accuracy performance values and compared with the help of Descriptive statistics, boxplot, and Significant tests.
Comparison of Different Sets of Features: box-plots: Figure 3 provides the performance value i.e., AUC, F-Measure, and accuracy of the models trained using selected sets of features and all features. We can see that the models developed using CCRA and AF have slightly better performance as compared to other techniques. The models developed using CCRA achieve 0.65 average AUC value, 0.98 max auc, and 0.78 Q3 AUC i.e., 25% models developed using CCRA have 0.78 AUC value. We can also observed that the models developed using AF have similar performance, but the number of features is more as compared to CCRA features sets.

Comparison of Different Sets of Features: Significant Test: In this study, the Wilcoxon signed-rank test is also applied to the AUC, F-Measure, and accuracy for statistically comparing the ability to predict the appropriate severity level of developed models by considering different sets of features an input. The objective of this testing is to find whether the performance of the models depends on input sets of features or not. The considered null hypothesis for this paper is "the defect severity level prediction models developed by considering different sets of features as an input are significantly same". The considered null hypothesis is only accepted if the obtained p-values using Wilcoxon signed-rank test is greater than 0.05. The results of Wilcoxon signed-rank test are depicted in Table IV. We can see that the models developed using all features, significant sets of features, and uncorrelated sets of features are significantly same.

TABLE IV: Significant tests: Different Sets of Features

	AF	SIGF	CCRA	PCA
AF	0	0	0	1
SIGF	0	0	0	1
CCRA	0	0	0	1
PCA	1	1	1	0

D. *Classification Techniques*

The predictive ability of developed defect severity level prediction models using different classification techniques are computed using AUC, F-Measure, and accuracy performance values and compared with the help of Descriptive statistics, box-plot, and Significant tests. In this work, we have used eleven different classification techniques such as multinomial naive bayes (MNB), bernoulli naive bayes (BNB), gaussian

TABLE V: Significant tests: Classification Techniques

	MNB	BNB	GNB	LOGR	DST	SVML	SVMP	SVMR	NNLBFG	NNSGD	NNADAM
MNB	0	1	1	1	1	1	1	1	1	0	1
BNB	1	0	1	1	1	1	1	1	1	0	1
GNB	1	1	0	0	1	0	1	1	0	1	1
LOGR	1	1	0	0	1	0	1	1	0	1	0
DST	1	1	1	1	0	1	1	0	1	1	1
SVML	1	1	0	0	1	0	1	1	0	1	0
SVMP	1	1	1	1	1	1	0	1	1	1	1
SVMR	1	1	1	1	0	1	1	0	1	1	1
NNLBFG	1	1	0	0	1	0	1	1	0	1	0
NNSGD	0	0	1	1	1	1	1	1	1	0	1
NNADAM	1	1	1	0	1	0	1	1	0	1	0

naive bayes (GNB), Logistic Regression (LOGR), decision tree (DST), SVM with linear kernel (SVML), SVM with polynomial kernel (SVMP), SVM with RBF kernel (SVMR), Neural network with LBFG (NNLBFG), Neural network with SGD (NNSGD), and Neural network with ADAM (NNADAM) with 5-fold cross-validation to train defect severity level prediction models.

Comparison of Classification Techniques: Descriptive Statistics and box-plots: Figure 5 provides the performance value, i.e., AUC, F-Measure, and accuracy of different classifiers in terms of Box-Plot diagrams and descriptive statistics. It is clear from Figure 2 that the models trained using SVM with polynomial kernel have better predictive ability to predict the appropriate severity level to the defects present in the bugs reports as compared to other models. The models developed using SVM with polynomial kernel achieve 0.73 average AUC value, 0.98 max auc, and 0.89 Q3 AUC i.e., 25% models developed using SVM with polynomial kernel have 0.89 AUC value. However, the models developed using bernoulli naive bayes (BNB) have low predictive ability as compared to other techniques.

Comparison of Classification Techniques: Significant Test: In this study, the Wilcoxon signed-rank test is also applied to the AUC, F-Measure, and accuracy for statistically comparing the ability to predict the appropriate severity level of developed models using different classifiers. The objective of this testing is to find whether the models trained using different classification techniques have a significant improvement or not. The considered null hypothesis for this paper is "the defect severity level prediction models trained using different classifiers are significantly same". The considered null hypothesis is only accepted if the obtained p-values using Wilcoxon signed-rank test is greater than 0.05. The results of Wilcoxon signed-rank test on different pairs of classifiers are depicted in Table V. For the purpose of simplicity, we have used only two number for representing results, i.e., 0 means hypothesis accepted (models are significantly same) and 1 means hypothesis rejected (models are significantly different). While comparing the values present in Table V, we can observed that the models trained using different classifiers are significantly different for most of the cases.

VII. Conclusion

In this paper, we build a model to predict proper severity level of the defects using defect description. Different from existed researches, this work focus on seven different word embedding methods to represent the word not just as a number but as a vector in n-dimensional space. The predictive ability of these methods are evaluated using three sets of features selected using feature selection techniques, and eleven different classifiers with 5-fold cross-validation. We have also used SMOTE techniques in order to handle the class imbalance problem. Finally, the predictive ability of these models are computed and compared using AUC, F-Measure, and accuracy performance values. Our main conclusions are the following:

- The high value of AUC confirms that the developed models using word embedding on balanced data have

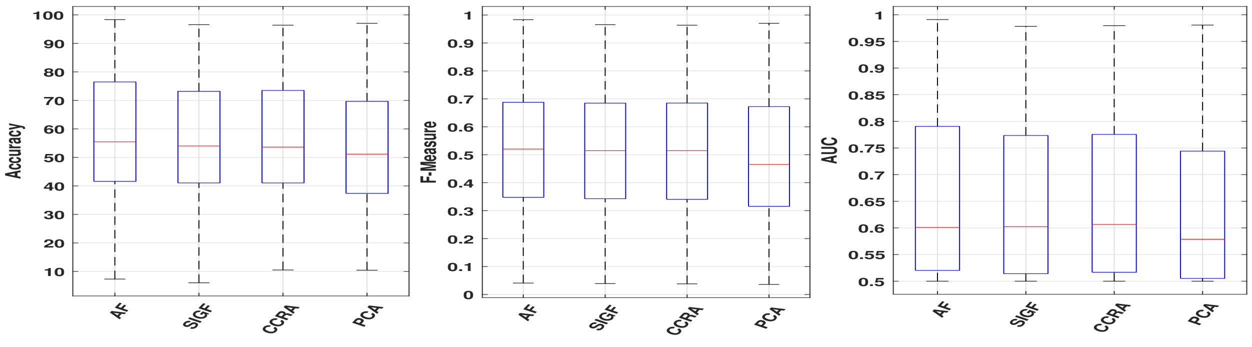


Fig. 3: Performance Box-Plot Diagram: Performance of Different Sets of Features

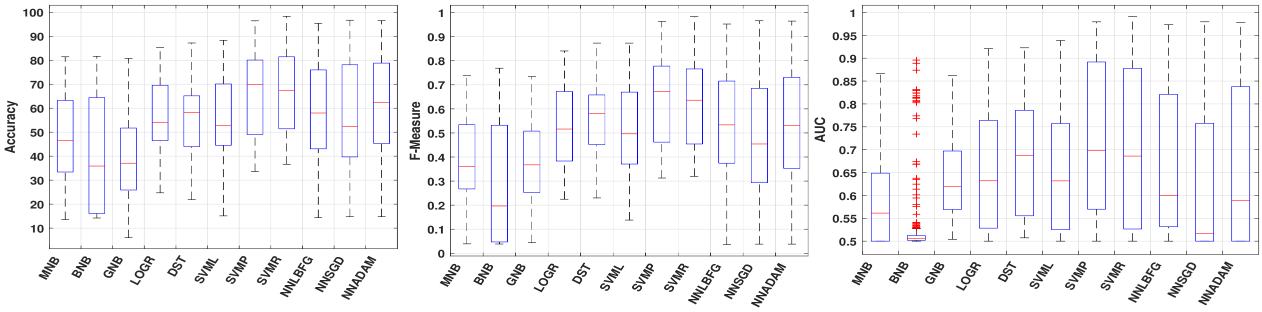


Fig. 5: Performance Box-Plot Diagram: Performance of Different Classification Techniques

the ability to predict severity levels of the defects present based on defect descriptions.

- The models developed by considered word vector computed using GLOVE and w2v have a better predictive ability as compared to other models.
- The defected severity levels prediction models developed using different word embedding methods are significantly different.
- The models trained on sampled data have significant improvement in predicting defect severity levels.
- The predictive ability of the models developed using significant uncorrelated features has a better ability to predict severity level as compared to all features.
- The models developed using SVM with polynomial kernel achieve significantly better performance as compared to other techniques.

In this study, developed are trained using most frequently used classifiers. Future work can be extended to deep-learning approach to achieve higher accuracy of software severity level prediction.

VIII. Acknowledgements

This research is funded by TestAIng Solutions Pvt. Ltd.

References

[1] R. Malhotra and A. Jain, "Fault prediction using statistical and machine learning methods for improving software quality," *Journal of Information Processing Systems*.

[2] L. Kumar, S. Misra, and S. K. Rath, "An empirical analysis of the effectiveness of software metrics and fault prediction model for identifying faulty classes," *Computer Standards & Interfaces*, vol. 53, pp. 1–32, 2017.

[3] R. Malhotra, N. Kapoor, R. Jain, and S. Biyani, "Severity assessment of software defect reports using text classification," *International Journal of Computer Applications*, vol. 83, no. 11, 2013.

[4] G. Abaei, A. Selamat, and H. Fujita, "An empirical study based on semi-supervised hybrid self-organizing map for software fault prediction," *Knowledge-Based Systems*.

[5] S. Kim and E. J. Whitehead Jr, "How long did it take to fix bugs?" in *Proceedings of the 2006 international workshop on Mining software repositories*, 2006, pp. 173–174.

[6] P. Bhattacharya and I. Neamtiu, "Bug-fix time prediction models: can we do better?" in *Proceedings of the 8th Working Conference on Mining Software Repositories*, 2011, pp. 207–210.

[7] A. More and D. P. Rana, "Review of random forest classification techniques to resolve data imbalance," in *2017 1st International Conference on Intelligent Systems and Information Management (ICISIM)*. IEEE, 2017, pp. 72–78.

[8] N. Junsomboon and T. Phienthrakul, "Combining over-sampling and under-sampling techniques for imbalance dataset," in *Proceedings of the 9th International Conference on Machine Learning and Computing*, 2017, pp. 243–247.

[9] T. Menzies and A. Marcus, "Automated severity assessment of software defect reports," in *2008 IEEE International Conference on Software Maintenance*. IEEE, 2008, pp. 346–355.

[10] R. Jindal, R. Malhotra, and A. Jain, "Software defect prediction using neural networks," in *Proceedings of 3rd International Conference on Reliability, Infocom Technologies and Optimization*. IEEE, 2014, pp. 1–6.

[11] S. Ghaluh Indah Permata, "An attribute selection for severity level determination according to the support vector machine classification result," in *proceedings intl conf information system business competitiveness*, 2012.

[12] M. Sharma, P. Bedi, K. Chaturvedi, and V. Singh, "Predicting the priority of a reported bug using machine learning techniques and cross project validation," in *2012 12th International Conference on Intelligent Systems Design and Applications (ISDA)*. IEEE, 2012, pp. 539–545.

[13] M. Gayathri and A. Sudha, "Software defect prediction system using multilayer perceptron neural network with data mining," *International Journal of Recent Technology and Engineering*, vol. 3, no. 2, pp. 54–59, 2014.

[14] D. L. Gupta and K. Saxena, "Software bug prediction using object-oriented metrics," *Sādhanā*, vol. 42, no. 5, pp. 655–669, 2017.

[15] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[16] N. V. Chawla, "Data mining for imbalanced datasets: An overview," in *Data mining and knowledge discovery handbook*. Springer, 2005, pp. 853–867.

[17] T. R. Hoens and N. V. Chawla, "Imbalanced datasets: from sampling to classifiers," *Imbalanced Learning: Foundations, Algorithms, and Applications*, pp. 43–59, 2013.

[18] S. S. Rathore and S. Kumar, "An empirical study of some software fault prediction techniques for the number of faults prediction," *Soft Computing*, vol. 21, no. 24, pp. 7417–7434, 2017.

[19] R. Malhotra, *Empirical research in software engineering: concepts, analysis, and applications*. CRC Press, 2016.

DOI: 10.15439/2021F98
ISSN 2300-5963 ACSIS, Vol. 25

Modeling and Verification using Different Notations for CPSs: The One-Water-Tank Case Study

Sergey Staroletov
Polzunov Altai State Technical University /
Institute of Automation and Electrometry, Russia
Email: serg_soft@mail.ru

Horst Schulte, Thomas Baar
HTW Berlin, School of Engineering I,
Berlin, Germany
Email: {schulte, baar}@htw-berlin.de

Ivan Konyukhov, Nikolay Shilov
Innopolis University, Innopolis,
Republic of Tatarstan, Russia
Email: i.konyukhov@innopolis.ru, shiloviis@mail.ru

Andrei Rozov, Tatiana Liakh, Vladimir Zyubin
Institute of Automation and Electrometry / NSU,
Novosibirsk, Russia
Email: {rozov, liakh, zyubin}@iae.nsk.su

Abstract—**The choice of an adequate notation and subsequent system formalization are the crucial points for the design of cyber-physical systems (CPSs). Here, an appropriate notation allows an explicit specification of the deterministic system behavior for specified initial states and inputs. We base our study on an industrial example (water tank) that comprises nominal as well as safety-critical states, and focus on the notation's support to validate/verify crucial safety properties. Several industrial notations (e.g. Matlab/Simulink©) to design and simulate such a hybrid system have been tried based on our physical model. In addition, we remodel our example using the well-founded mathematical formalism of hybrid automata. It enables us to formally express and verify important safety properties using the theorem prover KeYmaera.**

I. Introduction

THE increasing complexity and use of cyber-physical systems (embedded in particular) in our lives requires a reassessment of the system design principles and tools. The most challenging designs are safety-critical systems, such as transportation systems (e.g., airplanes, cars, and trains), infrastructure systems (power grid, water management, traffic control), and medical equipment. Here, the correct behavior under different environmental conditions must be ensured. The system must be fault-tolerant, i.e. in the case of faults, it must be automatically responsive to prevent worse [1].

In practice, different mathematical and formal models are used for system design and analysis, validation and verification. In the modeling process, a distinction should be made between two kinds of models [2]. The models of the first type are used for simulation and should be able to represent the characteristic behavior of systems in the physical world; the value of a model of this type lies in how well its properties match those of the real world, but we should emphasize that the model fidelity is never perfect.

Models of the second type serve as blueprints or specifications for the design of real-world (control) systems; system design and implementation follow the specification of a model; thus, the value of a model of the second type lies in how well and easy a real system can be constructed to implement the behavior specified in the model.

Due to the complexity and heterogeneity, mathematical models of real-world control systems should be described by different mixed representations. Hybrid systems are a particular class of mixed models that focus on the combination of discrete and continuous subsystems. For example, controllers for local operating regions can be represented mathematically by continuous-time systems, where the switching mechanism between different control regions are mostly represented as discrete event systems. The whole behavior is described by a hybrid model including event-based, discrete state changes, and continuous property changes over time.

Because of space limitations, a survey of a vast educational and periodic literature on simulation approach to design, modeling and validating real-world cyber-physical systems is out of the scope of this short conference paper (interested reader can follow our survey [3]). An overview on the formalism and notation of hybrid systems is available in [4] and [5]. Topics related to hybrid systems model-checking-oriented specification and verification are addressed in many papers, e.g. [6], [7], and [8]. An introduction to proof-oriented logical analysis and verification of hybrid systems is presented in the monographs [9], [10].

There are also some publications related to model checking and verification of particular hybrid systems. For example, paper [11] discusses hybrid system modeling and control for large-scale power systems; an analysis of embedded control software in safety-critical systems like autonomous vehicles in urban environments is presented in [12].

The rest of the paper is organized as follows. The one-tank system as our benchmark is informally introduced in Section II; in addition, this section presents a mathematical analysis of the system. In Section III, we use prevailing industrial techniques to ensure model properties, including safety-critical behavior. An alternative approach is given in

This work was supported by the German Science Foundation (DFG-Deutsche Forschungsgemeinschaft), funding program for international research cooperation, grant no. BA 6655/1-1, SCHU3174/3-1.

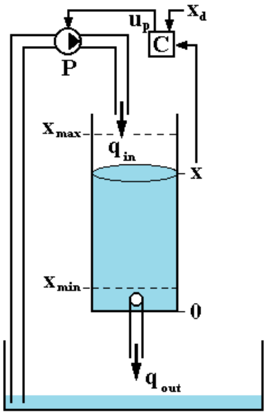


Fig. 1. The water tank system

Symbol	Description	Value	Unit
x	water level of the tank	$\in [0, H]$	m
x_d	desired water tank level (set point)	$\in [x_{\min}, x_{\max}]$	m
u_p	plant input / controller output	$\in [u_{p,\min}, u_{p,\max}]$	V
q_{in}	incoming flow rate	$\in [0, q_{\max}]$	m^3/s
A_T	cross-section of the tank	$7.9 \cdot 10^{-3}$	m^2
A_{out}	cross-section of the output orifices	$2.9106 \cdot 10^{-5}$	m^2
K_{pum}	pump coefficient	$8.374 \cdot 10^{-6}$	$m^3/(Vs)$
$x_{\min}$	lowest permitted water level	0.1	m
$x_{\max}$	highest permitted water level	0.5	m
H	tank height	0.65	m
$q_{\max}$	maximum flow rate of pump P	$100 \cdot 10^{-6}$	m^3/s
α_{out}	flow coefficient of output orifices	0.7	–
g	Earth gravity	9.81	m/s^2

TABLE I
VARIABLES AND PARAMETERS OF THE WATER TANK SYSTEM

Section IV with the formalization of the benchmark using the notation of hybrid automaton, which allows the rigorous logical verification of safety properties using the proof assistant KeYmaera. Finally, Section V concludes the paper.

II. Process Control Benchmark: Water tank

The water tank system (Fig. 1) can be viewed as a prototype (a core in some sense) of many industrial process control applications, e.g. in chemical plants or oil and gas systems. The typical control problem is to track the tank level by an input flow q_{in} under various disturbances. Moreover, the water tank process with one, two or three tanks is often used as a benchmark for fault diagnosis and isolation as well as fault-tolerant control [13].

The system consists of the water tank with cross-sectional area A_T and height H, orifice with cross-sectional area A_{out}, level sensor, pumping unit P, controller C and water basin.
In this setup, the pump provides in-feed of the water q_{in} to the tank, and the outflow of the tank is denoted as q_{out}.

The following conditions with regard to the system are used to describe the level of the water x in the tank:

- The level of the water is measured by the sensor.
- The controller is able to force on the pumping unit by changing the voltage u_p applied to the input terminals of the pump.
- In nominal conditions, the controller allows to keep desired level of the water x_d in a predicted range $[x_{min}, x_{max}]$.
- The controller handles the situation of low level protection (when $x < x_{min} + \Delta_{IN}$) as well as the situation of the high level protection (when $x > x_{max} - \Delta_{IN}$).
- The level protection states are entered well before water the level x gets too close to x_{min}, x_{max}.

A. Model-based Hybrid Controller Design

The dynamic equation for the water level is derived as follows. The rate change of the water level in a time is given by

$$\dot{x}(t) = \frac{1}{A_T}(q_{in}(t) - q_{out}(t)), \qquad (1)$$

where x, A_T,q_{in}, q_{out} are the water level, cross-sectional area of the tank, inflow rate, outflow rate, respectively. Next, note that the inflow rate to the tank is given by

$$q_{in}(t) = K_{pum} u_p(t), \qquad (2)$$

where K_{pum} is the pump coefficient and $u_p(t)$ is the voltage applied to the pump. In addition, using the Torricelli's law for a flow through a small orifice, the outflow rate of the water from the tank is given by

$$q_{out}(t) = \alpha_{out} A_{out} \sqrt{2gx(t)}, \qquad (3)$$

where g is the gravitational acceleration, A_{out} denotes the cross-sectional area of the orifice and α_{out} is the flow coefficient of the orifice. Using the (1–3), we obtain the dynamic equation for the water level in the tank as

$$\dot{x}(t) = \frac{1}{A_T}(-\alpha_{out} A_{out} \sqrt{2gx(t)} + K_{pum} u_p(t)) \qquad (4)$$

or in simple notation,

$$\dot{x}(t) = -\gamma \sqrt{x(t)} + \beta u_p(t) \ , \qquad (5)$$

where

$$\gamma := \frac{\alpha_{out} A_{out}}{A_T} \sqrt{2g} \quad , \quad \beta := \frac{K_{pum}}{A_T} \quad . \qquad (6)$$

All variables and values of actual parameters are recorded in Table I.

B. Hybrid Control for Low and High Level Protection

Due to the linear controller design for the nonlinear water tank system (that operates in a large operating range $x \in [x_{min}, x_{max}]$) and additional model uncertainties, the desired reference model with an overshoot free behavior is not exactly reached. Nevertheless, in order to be able to fulfill the requirements the system is extended by two control states. To distinguish them from the nominal states, these are denoted by *High Level Protection* (HP) and *Low Level Protection* (LP) state. During HP the controller output is set to $u_p = u_{p,min} = 0$. That means the pump is switched off as long as condition $x > x_{max} - \Delta_{OUT}$ holds. On the other hand, in the case of the LP state, the pump is set to the maximum flow rate q_{max}, that means (see (2)) the controller output is set to the constant value $u_p = u_{p,max}$. Note that the conditions from the nominal control state to the LP/HP state and from the LP/HP state to the nominal control state contain different delta values Δ_{IN} and Δ_{OUT}. With this, if $\Delta_{IN} < \Delta_{OUT}$ is fulfilled, it can be guaranteed that fast switching between states (scattering) is avoided.

III. Quality Assurance by Simulation

The so-called hardware (HIL) and software-in-the-loop (SIL) approaches are popular in industrial practice to ensure code quality. The difference between HIL and SIL is that the latter does not use target hardware to verify the code. In this paper, we use SIL-based verification, where the plant model and the controller code are simulated in the same environment.

For the verification by simulation with Matlab/Simulink©, the following controller settings for the nominal controller, the calculation rule of the controller coefficients and the switching conditions LP/HP are chosen as follows.

- **Nominal Controller Design**: We obtain the controller coefficients

$$k_p = 31.2634\,, \qquad k_I = 0.4016 \tag{7}$$

  by using a desired reference dynamics with $\tau_{ref} = 30$, a chosen steady-state water level of $x^{ss} = 0.2\,\mathrm{m}$ and the given plant parameter of Table I.
- **Switching conditions**: The thresholds of the lowest and highest permitted water level are determined by the controller requirements. The values are listed in Table I. The relative thresholds $\Delta_{IN,OUT}$ are defined as

$$\Delta_{IN} = 0.01\,, \qquad \Delta_{OUT} = 0.05\,. \tag{8}$$

A selected simulation result for a given curve of reference values $x_d(t)$, $t = [0, 2000s]$ using the parameter setting (7), (8) is shown in Fig. 2. In the bottom diagram, the state values correspond to the implementation with the *high level protection* state denoted as STATE_HP = 1, the *nominal control* state as STATE_NC = 2, and the *low level protection* state as STATE_LP = 3. The simulation results clearly show (by visual inspection) that the controller meets the previously defined requirements for the given reference case.

To address new challenges of today's control software, some of the authors proposed a *process-oriented approach*, which has been implemented in a family of domain-specific programming languages such as Reflex, Industrial C and PoST. In this approach, control software is represented as a set of interacting processes, where processes are state machines enhanced with special operators that implement concurrent flow control and time-interval managing [14]. So, after modeling, these languages can be used for implementation of the system.

IV. Formal Verification with KeYmaera

In the previous sections, the water tank system has been formalized using different notations. Remark that all industrial notations discussed above use exclusively simulation as a technique to check, whether the system behaves as expected. Generally, there is a lack of tool support for proving system properties merely based on the static system description. Only a rigorous analysis of the system can certify, that all expected safety properties actually hold under all circumstances.

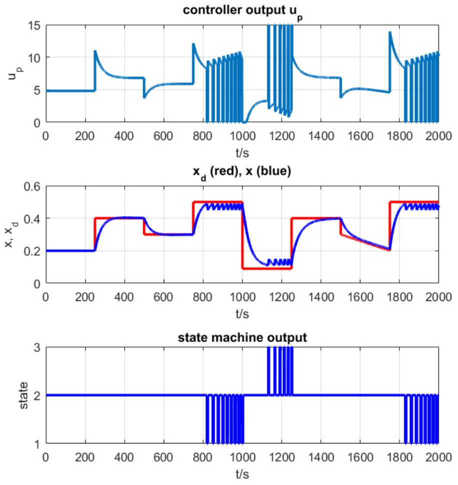


Fig. 2. Verification by simulation with Matlab/Simulink©

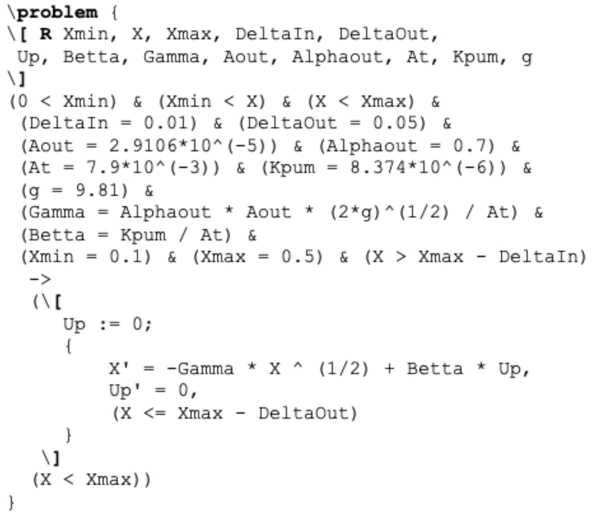
Because the simulation lacks soundness and comprehensiveness, we develop an additional system formalization using the hybrid automata approach [15], [4]. In this paper, we focus on the formal verification of the safety property that the current water level x never exceeds x_{max}, i.e. always $x < x_{max}$ holds.

As a verification tool we have chosen and used KeYmaera [16], [9]. We encoded a part of the problem in the special input code that is presented in Fig. 3. Actually, we have excerpted a part of the complete system description and focused on the state HP, which is most relevant for safety property $x < x_{max}$.

A hybrid automaton consists of states connected by transitions that encode the control flow. In contrast to classical automata, a state (e.g. HP) can be annotated with differential equations[1] (e.g. $x' = -\gamma\sqrt{x} + \beta u_p$, $u_p' = 0$) to encode, how the values of continuous variables (e.g. x, u_p) evolve while the system remains in the state. In addition, a state can be annotated with so-called domain constraints. A domain constraint is a condition to be true while the system is within the annotated state (e.g. $x >= x_{max} - \Delta_{OUT}$ for state HP); when some domain constraint of a state becomes invalid because of the change of some values of variables (e.g. value for x has fallen below $x_{max} - \Delta_{OUT}$), then the system is forced to leave the state via an outgoing transition (in case of HP, it can be left to state NC or to the final state). Note that a state can be left at any random time as long as there is an outgoing transition whose annotation condition allows its firing.

In order to check our safety property $x < x_{max}$, it is sufficient to show the property for the final state, since all states of the automaton (HP, NC, LP) are directly connected with the final state by an unconditioned transition that can fire any time and move the system to the final state. Thus, it is sufficient

[1] We use here x' instead of $\dot{x}$ to denote the derivative of x simply for consistency with the literature, e.g. [10].



```
\problem {
\[ R Xmin, X, Xmax, DeltaIn, DeltaOut,
 Up, Betta, Gamma, Aout, Alphaout, At, Kpum, g
\]
(0 < Xmin) & (Xmin < X) & (X < Xmax) &
 (DeltaIn = 0.01) & (DeltaOut = 0.05) &
 (Aout = 2.9106*10^(-5)) & (Alphaout = 0.7) &
 (At = 7.9*10^(-3)) & (Kpum = 8.374*10^(-6)) &
 (g = 9.81) &
 (Gamma = Alphaout * Aout * (2*g)^(1/2) / At) &
 (Betta = Kpum / At) &
 (Xmin = 0.1) & (Xmax = 0.5) & (X > Xmax - DeltaIn)
  ->
  (\[
     Up := 0;
     {
          X' = -Gamma * X ^ (1/2) + Betta * Up,
          Up' = 0,
          (X <= Xmax - DeltaOut)
     }
   \]
  (X < Xmax))
}
```

Fig. 3. HP in KeYmaera notation

to specify the safety property $x < x_{max}$ as part of the post-condition.

The only state that could violate safety property $x < x_{max}$ is *HP*, since for all other states this property is a weaker form of the state's domain constraint.

For state *HP*, the argumentation goes as follows:

- *Before entering HP, the value for voltage u_p is set to 0 and while the system stays in HP, this value remains 0 since $u'_p = 0$ holds in HP.*
- *Furthermore, upon entering HP, we know that $x < x_{max}$ as it is specified on all incoming transitions. Since we have $x' = -\gamma\sqrt{x} + \beta u_p$ together with $u_p = 0$, we know that x' is always negative in HP and x falls over time.*
- *Consequently, $x < x_{max}$ does hold when the process enters the state HP and for the whole period of staying in HP.*

The code for *HP* in KeYmaera notation is shown in Fig. 3.

The KeYmaera tool allows to transform the above informal mathematical argumentation into a formal proof and does check every proof step for correctness. At the end we get a formal verification that the desired safety property actually holds. Note that the verification process helps to make all assumptions explicit, e.g. it is very crucial to know that $0 < \Delta_{IN}$.

V. Conclusion

There are many different notations for the design, modeling, and analysis of control systems. They have proven to be useful in numerous industrial projects but also differ in terms of the used paradigms (e.g. object orientation, explicit state representation, etc.).

However, as our example shows, there is still a considerable gap in the usage of modeling concepts that still prevents an easy translation of, for example, Matlab/Simulink© models into input models for hybrid theorem provers like KeYmaera. It should also be noted that the KeYmaera tool cannot automatically find proofs for non-trivial properties and requires substantial user input (cmp. [17]).

In future, we plan to address these problems. We are developing an intermediate notation between industrial-strong modeling notations and hybrid automata to gain synergy. Our intermediate notation will semantically be strongly based on mathematically well-founded hybrid automata, but will also provide higher modeling concepts, that will facilitate the transition of industry models into our notation.

References

[1] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault-Tolerant Control*, 2nd ed. Springer-Verlag Berlin Heidelberg, 2006. ISBN 978-3-540-35653-0

[2] E. A. Lee, "Fundamental limits of cyber-physical systems modeling," *ACM Transactions on Cyber-Physical Systems*, vol. 1, no. 1, 2016. doi: 10.1145/2912149

[3] S. Staroletov, N. Shilov, I. Konyukhov, V. Zyubin, T. Liakh, A. Rozov, I. Shilov, T. Baar, and H. Schulte, "Model-driven methods to design of reliable multiagent cyber-physical systems," in *CEUR Workshop Proceedings*, vol. 2478, 2019, pp. 74–91.

[4] T. A. Henzinger, "The theory of hybrid automata," in *Proceedings, 11th Annual IEEE Symposium on Logic in Computer Science, New Brunswick, New Jersey, USA, July 27-30, 1996*, 1996. doi: 10.1109/LICS.1996.561342 pp. 278–292.

[5] R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, Eds., *Hybrid Systems*, ser. Lecture Notes in Computer Science, vol. 736. Springer, 1993. doi: 10.1007/3-540-57318-6

[6] R. Alur, T. A. Henzinger, and P. H. Ho, "Automatic symbolic verification of embedded systems," in *Proc. of the 14th Annual Real-time Systems Symp.*, 1993. doi: 10.1109/32.489079 pp. 2–11.

[7] B. I. Silva, O. Stursberg, B. H. Krogh, and S. Engell, "An assessment of the current status of algorithmic approaches to the verification of hybrid systems," in *Proc. of the 40th IEEE Conf. on Decision and Control*, 2001. doi: 10.1109/CDC.2001.980711 pp. 2867–2874.

[8] S. Mitsch and A. Platzer, "Modelplex: verified runtime validation of verified cyber-physical system models," *Formal Methods System Design*, vol. 49, no. 1,2, pp. 33–74, 2016. doi: 10.1007/s10703-016-0241-z

[9] A. Platzer, *Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics*. Heidelberg: Springer, 2010. ISBN 978-3-642-14508-7

[10] A. Platzer, *Logical Foundations of Cyber-Physical Systems*. Springer, 2018. ISBN 978-3-319-63587-3

[11] G. K. Fourlas, K. J. Kyriakopoulos, and C. D. Vournas, "Hybrid systems modeling for power systems," *Circuits and Systems Magazine, IEEE*, vol. 4, no. 3, pp. 16–23, 2004. doi: 10.1109/MCAS.2004.1337806

[12] K. G. Larsen, "Verification and performance analysis for embedded systems," in *Third IEEE International Symposium on Theoretical Aspects of Software Engineering (TASE)*. Tianjin, China: IEEE Computer Society, July 2009. doi: 10.1109/TASE.2009.66

[13] A. Kroll and H. Schulte, "Benchmark problems for nonlinear system identification and control using soft computing methods: Need and overview," *Applied Soft Computing*, vol. 25, no. 12, pp. 496–513, December 2014. doi: 10.1016/j.asoc.2014.08.034

[14] I. Anureev, N. Garanina, T. Liakh, A. Rozov, H. Schulte, and V. Zyubin, "Towards safe cyber-physical systems: the Reflex language and its transformational semantics," in *2019 International Siberian Conference on Control and Communications (SIBCON)*. IEEE, 2019. doi: 10.1109/SIBCON.2019.8729633 pp. 1–6.

[15] R. Alur, C. Courcoubetis, T. A. Henzinger, and P. Ho, "Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems," in *Hybrid Systems*, 1992. doi: 10.1007/3-540-57318-6 pp. 209–229.

[16] J.-D. Quesel, S. Mitsch, S. Loos, N. Aréchiga, and A. Platzer, "How to model and prove hybrid systems with KeYmaera: A tutorial on safety," *STTT*, vol. 18, no. 1, pp. 67–91, 2016. doi: 10.1007/s10009-015-0367-0

[17] T. Baar and S. Staroletov, "A control flow graph based approach to make the verification of cyber-physical systems using KeYmaera easier," *Modeling and Analysis of Information Systems*, vol. 25, no. 5, pp. 465–480, 2018. doi: 10.18255/1818-1015-2018-5-465-480

DOI: 10.15439/2021F46
ISSN 2300-5963 ACSIS, Vol. 25

Fed-agent – a Transparent ACID-Enabled Transactional Layer for Multidatabase Microservice Architectures

Lazar Nikolić
University of Novi Sad, Faculty of Technical Sciences
Email: lazar.nikolic@uns.ac.rs

Vladimir Dimitrieski
University of Novi Sad, Faculty of Technical Sciences
Email: dimitrieski@uns.ac.rs

***Abstract*—With the recent expansion of specialized databases and departure from the "one size fits all" paradigm, engineers might decide to use multiple databases. Each database holds a representation of a data object but offers transactions and consistency guarantees only locally. Existing solutions either require additional coding or do not provide global ACID transactions. In this paper, we present fed-agent, a transactional layer that provides global consistency and ACID transactions for single data objects within multidatabase systems. It requires no additional coding besides configuration files. We show that fed-agent scales linearly and introduces an overhead small enough for most microservice solutions.**

I. Introduction

Since the mid-2000s, the database community has shifted away from the "one size fits all" approach to database systems [1]. Many new specialized databases have emerged over the past decade, such as VoltDB [2] and Neo4j [3]. These databases are sometimes able to outperform relational databases by a large margin for their specialty workloads [4] [5]. An argument can be made that to be as performant as possible, a system should incorporate various types of databases. Data objects would be stored in different representations across multiple databases, but still logically represent a single entity. A blog post, for example, could have its content fully stored in a relational database. Its text content could be also stored in a text search engine, while viewership statistics calculated based on it could be stored in an analytical database. There needs to be a mechanism to synchronize data object representations and offer global consistency. It is important to note that databases usually cannot extend their consistency and transactional properties beyond their scope. For example, having only databases with Atomicity, Consistency, Isolation and Durability (ACID) transactions will not automatically make all distributed transactions ACID.

Traditionally, the two-phase commit (2PC) [6] protocol is used for data synchronization when high consistency is required, but at the cost of lower throughput [7]. Persistent message queues can be used instead, at the cost of less strict consistency [8], limiting the design space, increasing development costs, and harming the developer and user experience. Another usual approach is to implement Extract-Transform-Load (ETL) processes, usually as scripts that migrate data periodically between databases. ETL adds extra costs to implementing and maintaining migration scripts, which are subject to change on each schema update.

The microservice architecture is a widely used architecture for building cloud-based software solutions. Microservices expose an Application Programming Interface (API) that is used to access or manipulate data stored in databases. Each microservice knows how to transform request data into a persistent format used by a database. If most database operations are handled through API calls, the messages should contain enough information to deduce the state of the database.

In this paper, we present *fed-agent*, a transactional layer acting as a proxy that aims to provide global consistency and ACID guarantees for multidatabase systems. Other solutions either require additional coding or do not support ACID over multiple databases. Fed-agent does so for single data objects with no additional code needed besides configuration files. This way distributed transaction processing is facilitated transparently, allowing the engineers to focus on the business logic. In this paper we also prove that fed-agent can provide the above perks with a low overhead and linear scaling.

The rest of the paper is organized as follows. In Section II we present the architecture and core functionalities of fed-agent. The fed-agent evaluation results are given in Section III. In Section IV we discuss the related work, while in Section V we conclude the paper and discusses future work.

II. Architecture

Fed-agent acts as a layer that unifies read and write operations on a single data object over multiple databases. It communicates with databases through microservices built on top of them. This eliminates the need to write and maintain code that translates data between various representations, as mappers are already implemented as a part of microservices.

Fed-agent consists of multiple nodes within a single Raft [9] consensus group, with one leader and multiple followers. Only the leader can accept writes, while followers can serve read requests. In case of a leader's failure, one of the followers will become the leader so the fed-agent can continue accepting writes. We present the high-level overview of fed-agent architecture in Fig. 1. It shows a fed-agent cluster consisting of three nodes, with *fed-agent 1* being the leader. *Service 1* and *Service 2* are microservices with separate databases. A client sending writes to any service does so via *fed-agent 1,* while reads can be served by any node. For example, a read request can be routed to *fed-agent 3*.

Fed-agent identifies an operation and the data object being read or modified based on the HTTP request's body and URL. For example, *PUT "/users/123"* implies a user object with ID "123" is being updated with data from the body. Rules for extracting the information from a request are defined declaratively in a configuration file for each endpoint.

Fed-agent uses Multi-Version Concurrency Control (MVCC) [10]. Every write request on a data object creates a new version of the data object. Reads can only see committed data object versions based on its timestamp. If fed-agent detects a microservice response contains an uncommitted data object, it will replace it with the last committed object available for the request's timestamp. MVCC implementation ensures snapshot isolation level, leaving only anomalies that happen for predicate-based operations. Since fed-agent operates only on single data objects, these anomalies cannot happen and serializable isolation level is ensured.

III. Evaluation

To evaluate fed-agent, we created a setup that illustrates a typical scenario in practice. The setup consisted of a fed-agent cluster acting as a proxy for microservices connecting to a database. The microservices are *userservice*, using *PostgreSQL 13*, *textservice*, using *Elasticsearch 7.12.1*, and *geosvc* using *Tile38* [11]. Each microservice is written in the *Go* programming language and uses only low-level database drivers and the standard HTTP library. There are two data object types: *User* and *Tweet* types. The *User* type consists of four text fields: *id*, *handle, email*, and *password*. The *Tweet* type consists of three text fields: *id*, *userHandle*, and *content*, and two double-precision numbers representing *latitude* and *longitude*. For simplicity, we store full data objects in all databases. The services offer REST API with two operations: (i) access a single data object by id and (ii) upsert a data object. All services are developed as if used in isolation and contain no code for distributed transactions. Distributed transactions and coordination are covered by fed-agent configuration files.

We use *Yahoo Cloud Serving Benchmark* (YCSB) [12] workload in our benchmarks. YCSB consists of six workloads: Workload A (50/50 read/write), Workload B (95/5 read/write), Workload C (read-only), Workload D (read latest), Workload E (top 100 records), and Workload F (read-modify-write). We do not consider Workload E since fed-agent does not currently support scans. We also add *Load* workload consisting of 100% write operations.

Overheads are measured as a difference between latencies from direct API calls to a microservice and API calls via fed-agent proxy. For write (upsert) operations, all backend requests are done in parallel, so we measure the difference between the slowest microservice response and the fed-agent proxy response.

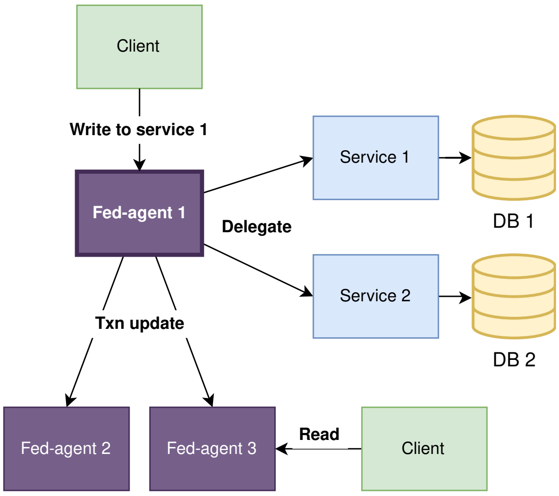


Fig 1. Fed-agent architecture overview

We acknowledge some threats to validity of this evaluation. Because all benchmarks were running on the public cloud and not on dedicated hardware, benchmarks results are subject to factors beyond our control, such as changes of network topology, network hiccups, or hardware failure. To minimize the effect, we ran each benchmark five times and reported the average. We also set the number of operations for each benchmark to 150000. Additionally, the bare-bones microservices used in benchmarks are developed by us for this sole purpose. Microservices in practice would have more complicated logic and would be running in much more complex architectures. Running these benchmarks in such a system might yield different results.

We run benchmarks to analyze how network overhead, payload size, number of fed-agent nodes, concurrency, and contention affect fed-agent. The results are discussed below.

Network overhead. This benchmark aims to measure the network overhead created by additional messages, so we ran it with a single client thread. We ran this benchmarks on an Amazon EC2 cluster of 10 *t1.micro* nodes. One node was acting as a client. Three nodes were running a fed-agent cluster. The three services connecting to distinct databases were each running on a different node. The databases were all running on their nodes. This benchmark considers all YCSB workloads. The results are shown in Table 1 and show that for a typical web-oriented microservice solution, one can expect about 7ms overhead for writes and about 1ms overhead for reads.

Payload size scaling. For this benchmark, we measure how much payload size can affect latency. Setup is identical to the one used for network overhead. YCSB Workload A is used to measure how payload size affects both reads and writes. Results are shown in Fig. 2.a. and Fig 2.b. Read overhead shows no clear correlation with payload size: increasing the payload size seems to yield results explained as usual

TABLE I.
AVERAGE OVERHEAD IN MILLISECONDS

Type	Workload A	Workload B	Workload C	Workload D	Workload F	Load
Read overhead (ms)	0.828	0.595	1.095	1.056	0.843	N/A
Write overhead (ms)	7.354	7.749	N/A	7.383	7.289	6.522

response time fluctuations. On the other side, write is heavily affected by the payload size. Overhead increases from about 10ms to 20ms for the first 100kB, which is the largest jump. Increasing the payload size from 100kB starts adding roughly 5ms per 100kB added. The main reason behind this increase is due to writes sending Raft messages and backend HTTP requests for every client request.

Node scaling. This benchmark measures the overhead of Raft messages as the number of nodes increases. Only writes are used because reads are strictly single node, meaning that only YCSB Load workload was considered. We ran the benchmarks on an Amazon EC2 cluster consisting of 10 *t1.micro* nodes all running fed-agent. We modified fed-agent to automatically commit all transactions because delegating HTTP requests is not affected by the number of fed-agent nodes. We are also only interested in the network overhead, so only a single client thread is used. The results are shown in Fig. 2.c. Since there is no need for consensus protocol for only one node, there is a spike in overhead when a second node is added, going from 0.4ms to 8ms. Each node added beyond the first adds 0.2ms-1ms latency. There is no indication this does not stay true for exceptionally large clusters, i.e., clusters of many tens or even hundreds of nodes.

Concurrency and contention. In the concurrency benchmark, we measured how overhead scales with an increasing number of concurrent users, but no contention. We define contention as the percentage of transactions accessing *hot data* that was 20% of total data objects. In the contention benchmark, we altered contention from 0% to 100% with a constant number of concurrent users. In both benchmarks, a concurrent user was represented by a thread running the YCSB Workload A benchmark. We capped the throughput at 3000 operations/second as to not overload the system that supported 5000 operations/second. The purpose of the benchmark was to measure the overhead of a usual workload and not to push the system to its limits. We ran the benchmarks on an Amazon EC2 cluster consisting of 7 *t3.2xlarge* nodes. Three fed-agent nodes, three *usersvc* instances using *PostgreSQL 13*, and a single client instance were deployed. *PostgreSQL* isolation level was set to serializable to match that of fed-agent, so we can more accurately compare the number of aborts.

Fig. 2.d. shows how the number of concurrent users affects the overhead. For this benchmark, we ran 100% read and 100% write workloads separately. The contention rate was set to 0%. Read overhead is mostly affected by fed-agent internal locking mechanism when fetching data object versions from the version chain. There appears to be a large spike going from 0.5s to 4s somewhere between 20 and 40 threads, which then plateaus before having a sharp drop to 0.9s at 100 threads. Our initial assumption is that at this point, disk I/O becomes the bottleneck, effectively masking network overhead of fed-agent. Write overhead is affected by the same locking mechanism when inserting versions to the chain, but also by the number of messages Raft transport can support, with the latter being the bigger factor.

Fig. 2.e. shows how contention rate affects the overhead, while Fig 2.f. shows how it affects the number of aborts. We set the number of threads to 40 for the benchmark. The overhead and the number of aborts appear largely unaffected by contention. The number of aborts is almost ten times higher than the baseline, which we attribute to distributed transactions simply being slower and causing more conflicts. The

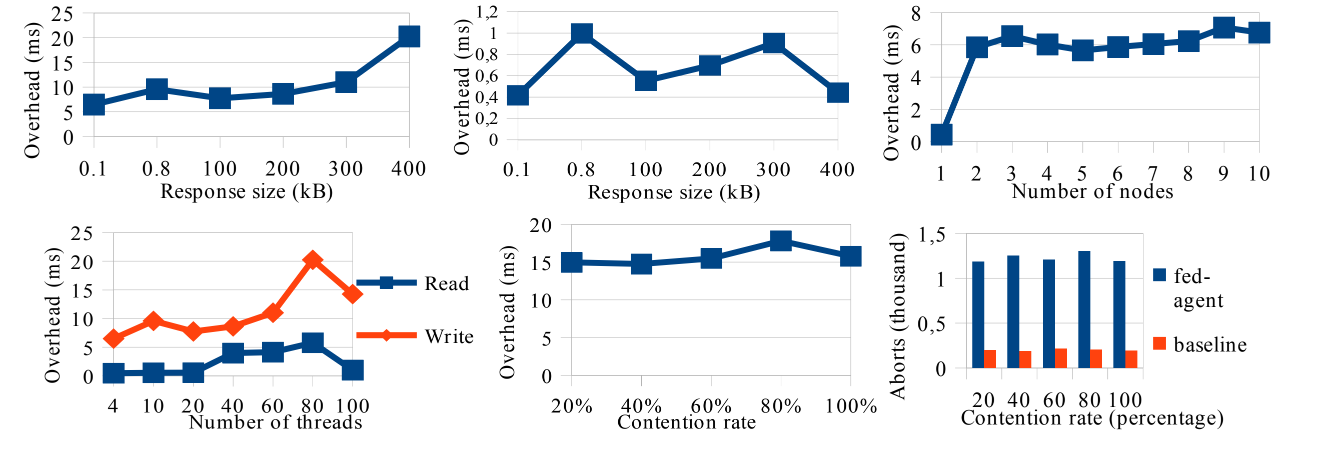


Fig 2. Benchmark results, left to right, top to bottom: a) payload size write scaling, b) payload size read scaling, c) write scaling with the number of nodes, d) concurrency scaling , e) contention scaling, f) aborts in relation to contention

total number of aborts is 1000-1200, which is small compared to the total number of operations, which is 150000.

Summary. The expected overhead for most use cases is 7-10ms for writes and 1ms for reads. For single-node fed-agent deployments, overhead for both reads and writes is less than 1ms. The highest measured overhead is 30ms for payload sizes of 400kB. This is an extreme scenario because most REST API payload sizes are 1-2kB [13]. In our opinion, fed-agent's overhead should be acceptable in microservice architectures with usual response times that are at least an order of magnitude higher, usually tens of milliseconds or longer. Fed-agent scales linearly: additional nodes add no overhead for reads and add 0.2ms-1ms overhead for writes.

IV. Related work

Distributed transactions in multidatabase systems are a well-known topic that can be traced back to the 1980s [6] [14]. It has seen some recent development, particularly in the domain of distributed databases and microservices.

ReTSO [15] has an architecture similar to fed-agent and serves as a global transaction tracker. Unlike ReTSO, fed-agent is an integrated solution that does not use any other components. GRIT [16] optimistically executes transactions, capturing write and read sets and then asynchronously applying them. It is not stated whether database service handling the write sets are an existing part of a microservice, or a component specifically developed for GRIT. Calvin [7] is conceptually similar to fed-agent but uses databases as backends instead of microservices. Deuteronomy [17] separates transactions from databases allowing it to execute them on multidatabase systems. Typhon [18] offers snapshot isolation level of access to single keys when accessed via Cerberus protocol, but not transparently. Dey et al [19] provides a Java client library for tracking transaction meta-data. The system heavily relies on test-and-set operations, limiting the choice of databases that can be used. None of the listed systems can automatically and transparently facilitate distributed transactions, but instead require using a client library.

V. Conclusion and future work

In this paper, we presented fed-agent, a transactional layer that provides ACID capabilities on single data objects over multidatabase microservice architectures. Each microservice can be developed to use a different database, regardless of consistency levels or ACID properties of the database. Fed-agent provides transparent transactions triggered automatically whenever an operation matching the provided configuration is detected. Microservices can be developed in isolation and require no code that implements distributed transactions. This brings the focus of engineers away from coordination into business logic. Our experiments show low overhead and linear scalability for a typical microservice setup.

There are several areas in which fed-agent can be improved in the future. First, we can allow users to declaratively define mappings between request types instead of being forced to use identical requests for all microservices. Second, the system can be split into shared-nothing partitions. Third, multi-object open transactions can be supported by introducing operations akin to SQL *BEGIN*, *COMMIT,* and *ABORT*. Fourth, operations reading ranges of data objects (scans) should also be supported as it is one of the common workloads described by the YCSB.

References

[1] M. Stonebraker, S. Madden, D. J. Abadi, S. Harizopoulos, N. Hachem, and P. Helland, "The end of an architectural era: it's time for a complete rewrite," Making Databases Work: the Pragmatic Wisdom of Michael Stonebraker, pp. 463–489, 2018.

[2] VoltDB, 10-May-2021. [Online]. Available: https://www.voltdb.com/. [Accessed: 23-May-2021].

[3] Neo4j Graph Database Platform, 13-May-2021. [Online]. Available: https://neo4j.com/. [Accessed: 23-May-2021].

[4] V. Gadepally, P. Chen, J. Duggan, A. Elmore, B. Haynes, J. Kepner, S. Madden, T. Mattson, and M. Stonebraker, "The BigDAWG polystore system and architecture," 2016 IEEE High Performance Extreme Computing Conference (HPEC), 2016.

[5] P. Bakkum and K. Skadron, "Accelerating SQL database operations on a GPU with CUDA," Proceedings of the 3rd Workshop on General-Purpose Computation on Graphics Processing Units - GPGPU '10, 2010.

[6] C. Mohan, B. Lindsay, and R. Obermarck, "Transaction management in the R* distributed database management system," ACM Transactions on Database Systems, vol. 11, no. 4, pp. 378–396, 1986.

[7] A. Thomson, T. Diamond, S.-C. Weng, K. Ren, P. Shao, and D. J. Abadi, "Calvin," Proceedings of the 2012 international conference on Management of Data - SIGMOD '12, 2012.

[8] W. Vogels, "Eventually consistent," Communications of the ACM, vol. 52, no. 1, pp. 40–44, 2009.

[9] D. Ongaro, and J. Ousterhout, „In search of an understandable consensus algorithm". In 2014 {USENIX} Annual Technical Conference ({USENIX}{ATC} 14) , pp. 305-319, 2014.

[10] P. A. Bernstein and N. Goodman, "Multiversion concurrency control —theory and algorithms," ACM Transactions on Database Systems, vol. 8, no. 4, pp. 465–483, 1983.

[11] Tile38. [Online]. Available: https://tile38.com/. [Accessed: 23-May-2021].

[12] B. F. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Sears, "Benchmarking cloud serving systems with YCSB," Proceedings of the 1st ACM symposium on Cloud computing - SoCC '10, 2010..

[13] C. Rodríguez, M. Baez, F. Daniel, F. Casati, J. C. Trabucco, L. Canali, and G. Percannella, "REST APIs: A Large-Scale Analysis of Compliance with Principles and Best Practices," Lecture Notes in Computer Science, pp. 21–39, 2016.

[14] Y. Breitbart, H. Garcia-Molina, and A. Silberschatz, "Overview of multidatabase transaction management," CASCON First Decade High Impact Papers on - CASCON '10, 2010.

[15] F. Junqueira, B. Reed, and M. Yabandeh, "Lock-free transactional support for large-scale storage systems," 2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops (DSN-W), 2011.

[16] G. Zhang, K. Ren, J.-S. Ahn, and S. Ben-Romdhane, "GRIT: Consistent Distributed Transactions Across Polyglot Microservices with Multiple Databases," 2019 IEEE 35th International Conference on Data Engineering (ICDE), 2019. Conference on Data Engineering (ICDE) (pp. 2024-2027). IEEE.

[17] Levandoski, J. Justin, D. Lomet, M. Mokbel and K. Zhao. "Deuteronomy: Transaction Support for Cloud Data." CIDR (2011).

[18] V. Arora, F. Nawab, D. Agrawal, and A. E. Abbadi, "Typhon: Consistency Semantics for Multi-Representation Data Processing," 2017 IEEE 10th International Conference on Cloud Computing (CLOUD), 2017.

[19] A. Dey, A. Fekete, and U. Rohm, "Scalable distributed transactions across heterogeneous stores," 2015 IEEE 31st International Conference on Data Engineering, 2015.

Joint 41$^{\text{st}}$ IEEE Software Engineering Workshop and 8$^{\text{th}}$ International Workshop on Cyber-Physical Systems

THE IEEE Software Engineering Workshop (SEW) is the oldest Software Engineering event in the world, dating back to 1969. The workshop was originally run as the NASA Software Engineering Workshop and focused on software engineering issues relevant to NASA and the space industry. After the 25$^{\text{th}}$ edition, it became the NASA/IEEE Software Engineering Workshop and expanded its remit to address many more areas of software engineering with emphasis on practical issues, industrial experience and case studies in addition to traditional technical papers. Since its 31$^{\text{st}}$ edition, it has been sponsored by IEEE and has continued to broaden its areas of interest.

One such extremely hot new area are Cyber-physical Systems (CPS), which encompass the investigation of approaches related to the development and use of modern software systems interfacing with real world and controlling their surroundings. CPS are physical and engineering systems closely integrated with their typically networked environment. Modern airplanes, automobiles, or medical devices are practically networks of computers. Sensors, robots, and intelligent devices are abundant. Human life depends on them. CPS systems transform how people interact with the physical world just like the Internet transformed how people interact with one another.

The joint workshop aims to bring together all those researchers with an interest in software engineering, both with CPS and broader focus. Traditionally, these workshops attract industrial and government practitioners and academics pursuing the advancement of software engineering principles, techniques and practices. This joint edition will also provide a forum for reporting on past experiences, for describing new and emerging results and approaches, and for exchanging ideas on best practice and future directions.

TOPICS

The workshop aims to bring together all those with an interest in software engineering. Traditionally, the workshop attracts industrial and government practitioners and academics pursuing the advancement of software engineering principles, techniques and practice. The workshop provides a forum for reporting on past experiences, for describing new and emerging results and approaches, and for exchanging ideas on best practice and future directions.

Topics of interest include, but are not limited to:

- Experiments and experience reports
- Software quality assurance and metrics
- Formal methods and formal approaches to software development
- Software engineering processes and process improvement
- Agile and lean methods
- Requirements engineering
- Software architectures
- Design methodologies
- Validation and verification
- Software maintenance, reuse, and legacy systems
- Agent-based software systems
- Self-managing systems
- New approaches to software engineering (e.g., search based software engineering)
- Software engineering issues in cyber-physical systems
- Real-time software engineering
- Safety assurance & certification
- Software security
- Embedded control systems and networks
- Software aspects of the Internet of Things
- Software engineering education, laboratories and pedagogy
- Software engineering for social media

TECHNICAL SESSION CHAIRS

- **Bowen, Jonathan,** Museophile Ltd., United Kingdom
- **Hinchey, Mike** (Lead Chair), Lero-the Irish Software Engineering Research Centre, Ireland
- **Szmuc, Tomasz,** AGH University of Science and Technology, Poland
- **Zalewski, Janusz,** Florida Gulf Coast University, United States

PROGRAM COMMITTEE

- **Ait Ameur, Yamine,** IRIT/INPT-ENSEEIHT, France
- **Cicirelli, Franco,** Dimes - Unical, France
- **Ehrenberger, Wolfgang,** University of Applied Science, Germany
- **Gomes, Luis,** Universidade NOVA de Lisboa, Portugal
- **Gracanin, Denis,** Virginia Tech, United States
- **Havelund, Klaus,** Jet Propulsion Laboratory, California Institute of Technology, United States
- **Hsiao, Michael,** Virginia Tech, United States
- **van-Katwijk, Jan,** TU Delft, The Netherlands

- **Trybus, Leszek,** Rzeszow University of Technology, Poland
- **Vardanega, Tullio,** University of Padua, Italy
- **Velev, Miroslav,** Aries Design Automation, United States

DOI: 10.15439/2021F126
ISSN 2300-5963 ACSIS, Vol. 25

PRET-ization of uRISC Core

Martin Košťál
Faculty of Electrical Engineering
Czech Technical University in Prague
Email: kostama7@fel.cvut.cz

Michal Sojka
Czech Institute of Informatics, Robotics and Cybernetics
Czech Technical University in Prague
Email: michal.sojka@cvut.cz

***Abstract*—Modern safety-critical embedded systems have to be time-deterministic to guarantee safety. One source of time-nondeterminism are interrupts. This paper shows how to mitigate their influence in the system on a commercially available processor IP (Codasip uRISC) can be modified to exhibit time-determinism in real-time workloads and isolate interrupts. We extend the processor with fine-grained multithreading and isolated interrupt handling to localize time-nondeterminism caused by interrupts. We show a comparison between original and extended processors on a selection of TACleBench benchmarks. For interrupt-driven workloads, ideal interrupt isolation is achieved. The proposed modification can be used on other in-order single-issue processors.**

I. Introduction

Nowadays, Commercial-Of-The-Shelf (COTS) edge computing platforms are used in real-time applications. Their designers have to design them carefully to tolerate the time-nondeterminism of such platforms. Moreover, COTS platforms are optimized for average performance, so real-time applications require significant over-provisioning to meet timing requirements, especially in the worst case.

The Worst-Case Execution Time (WCET) bounds are given by two factors: the program and its inputs and the underlying processing architecture [1]. In this paper, we focus on processing architecture.

Modern processor architectures use many techniques to increase performance, but these are usually not time-deterministic. To name a few: instruction-level parallelism in superscalar architectures, branch prediction and speculative execution, out-of-order execution, caching and complex memory hierarchy. The problem with all these techniques is that timing depends on a complex micro-architectural state, which is often held secret from users. As a result, the execution time of a piece of code is subject to variance known as jitter.

Many researchers investigate the possibility of having a completely time-deterministic computing architecture. They propose either to modify existing architectures by replacing time-nondeterministic components with their deterministic counterparts. For example lockable [2] and partitionable caches [3], scratchpad memories [4] or time-predictable branch predictor [5]. Another approach is to develop a time-deterministic CPU, commonly referred to as PREcision Timed (PRET) machine [6], from scratch. Examples of such processors are FlexPRET [7] and Patmos [8].

One source of time-nondeterminism in real-time applications running in COTS processors is interrupt handling. Interrupts are essential to I/O communication [9], [10] and their arrival time is often unpredictable [11].

In this paper, we focus on interrupt isolation so that the execution time of software threads not requiring the interrupts for their function is not affected. We demonstrate how a COTS processor Intellectual Property (IP) can be modified to provide multithreading with interrupt isolation to decrease execution time jitter of real-time tasks. While FlexPRET [7] implements a similar feature in a completely new architecture, we show how such a feature can be implemented by modifying an existing architecture (Codasip uRISC). Finally, we evaluate the proposed modification in terms of additional FPGA resource cost and jitter reduction, which is completely eliminated for interrupt independent tasks.

Section II describes background information, namely the PRET machine and Codasip uRISC processor. In Section III we introduce our design of PRET-like uRISC core, and in Section IV, we describe exact modifications of the core. Evaluation of our modifications based on the CPU simulator is provided in Section V, and we conclude in Section VI.

II. Background

This section describes the background information that serves as a basis for our work.

A. *PRET machine*

According to Lee et al. [12], [6] an abstract PRET machine is a machine where "Repeatable timing is more important and more achievable than predictable timing".

The authors argue that software control over timing is orders of magnitude coarser than hardware controlled timing. The software approach leads to unnecessary over-provisioning and does not allow for software and hardware independent safety certifications because the software always has to be tailored to a specific target computing platform. There are three distinguishing features that a PRET machine has:

- timing instructions
- hardware threads
- isolated interrupts

Timing instructions set and clear deadline for a task. At the beginning of task execution, a deadline is set and at the end of execution, the deadline is cleared. If the deadline is not

This work was supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 832011 (THERMAC).

cleared before its due time, an exception occurs, which can deal with the missed deadline.

Hardware threads eliminate pipeline bubbles caused by branching. The thread switching is implemented as fine-grained multithreading, which interleaves instructions from all threads in a round-robin fashion.

Lastly, interrupts isolation allows to assign interrupts to hardware threads. Interrupts are handled as streams of sporadic events.

B. uRISC

Codasip uRISC processor is a pipelined core, written in the CodAl architecture description language [13]. It is used mainly for technology demonstrations by Codasip and has gained popularity in academia [14], [15], [16]. The uRISC instruction set architecture (ISA) supports 46 instructions with an effective single-cycle latency. The processor architecture is 32bit wide, has 32 registers in a register file. Its pipeline has fetch, decode, execute, and writeback stages. As a modified Harvard architecture, it has separate interfaces to memory for instructions and data. The memories, as well as peripherals, are connected through the AHB3 lite bus.

III. Design

This section describes the design of our two extensions of the uRISC processor: 1) fine-grained multithreading and 2) thread independent interrupt handling.

The fine-grained multithreading helps reduce execution time jitter by effectively eliminating pipeline stalls (pipeline bubbles), which are induced by the elimination of pipeline hazards. It can be shown that fine-grained multithreading minimizes pipeline stalls by eliminating control hazards. Control hazards are eliminated because consecutive instructions do not depend on the result of previous n instructions. For a core with a four-stage pipeline, such as the uRISC, this condition is fulfilled by dispatching instructions from one thread every fourth cycle.

Issuing instructions from a thread every n^{th} cycle increases the minimal, average and worst-case execution time of tasks, but the execution time jitter is eliminated. In order to utilize the pipeline optimally, at least n threads have to be executed on a core. As shown in [12] n-thread fine-grained multithreading achieves minimal execution time jitter for real-time tasks without performance loss.

Interrupts are a source of uncertainty for the execution time of any task. It is not only the delay caused by interrupt service routine, but the changes to the state of the components such as caches or branch target buffer also affect execution time. We assume that real-world sources of interrupts can be modelled as sporadic events with a non-zero minimal time between two consecutive events. However, there could be more sources of interrupts, which can arrive at the same time. Traditional processors can deal with simultaneous interrupts in many ways. One is to use interrupt masking, which forbids servicing interrupts simultaneously; the other is nesting, which allows a higher-priority interrupt to preempt lower priority interrupt.

Isolation of interrupts removes the need for interrupt masking and nesting by assigning a single interrupt controller to a single thread. This way, tasks requiring interrupts for their functionality (e.g. I/O services) do not affect the execution of other tasks and, vice versa, are not affected by the execution of other tasks in other threads.

This work does not implement deadline instructions proposed in [12] because the same functionality can be implemented with a timer interrupt without changing the ISA.

A drawback of our design is the lack of atomic instructions in the uRISC ISA, limiting the options of inter-thread data access. We assume that every thread executes a separate program, and the programs do not communicate with each other. We plan to add atomic instructions later.

This design can be used for other in-order single-issue cores, which have one-cycle latency of all instructions. For cores with branch prediction and operand forwarding, the number of dependent instructions could be smaller than the number of pipeline stages but is never greater.

IV. Implementation

This section describes the specific implementation of PRET uRISC core and test platform, which is used for evaluation.

A. PRET extensions

The implementation of the fine-grained multithreading extends the number of simultaneously executed threads to four. Instructions from multiple threads are dispatched in a round-robin fashion, and the pipeline never contains two instructions from the same thread.

To support four threads, changes are made to the pipeline as shown in Fig. 1 by grey elements. A program counter is quadrupled, and the register file size is increased four times so that each thread has its own program counter and a portion of the register file. The whole pipeline keeps track of the thread associated with each instruction, further referenced as thread id. The fetch stage implements thread switching logic. Instructions are fetched based on a periodic round-robin schedule. Once the instruction is fetched, it is associated with a thread id. The decode stage is modified to decode operand register addresses based on the thread id so that every instruction accesses only a portion of the register file associated with its thread. It is not possible to address registers in the register file which are associated with other threads. Execute stage requires no modification; only passing of thread id is implemented. The writeback stage is modified to decode a return register address based on the thread id.

The uRISC has one interrupt signal and a fixed address for the interrupt service routine (ISR). We extend the interrupt signal to four signals, one for each thread.

The original uRISC core has four instructions related to interrupts: interrupt enable, disable, call and jump. Implementation of these instructions is modified so that interrupt enable only enables interrupts in the thread it is executed in. Likewise, interrupt disable instruction disables interrupts only for the thread it is executed in. Interrupt enable register is quadrupled.

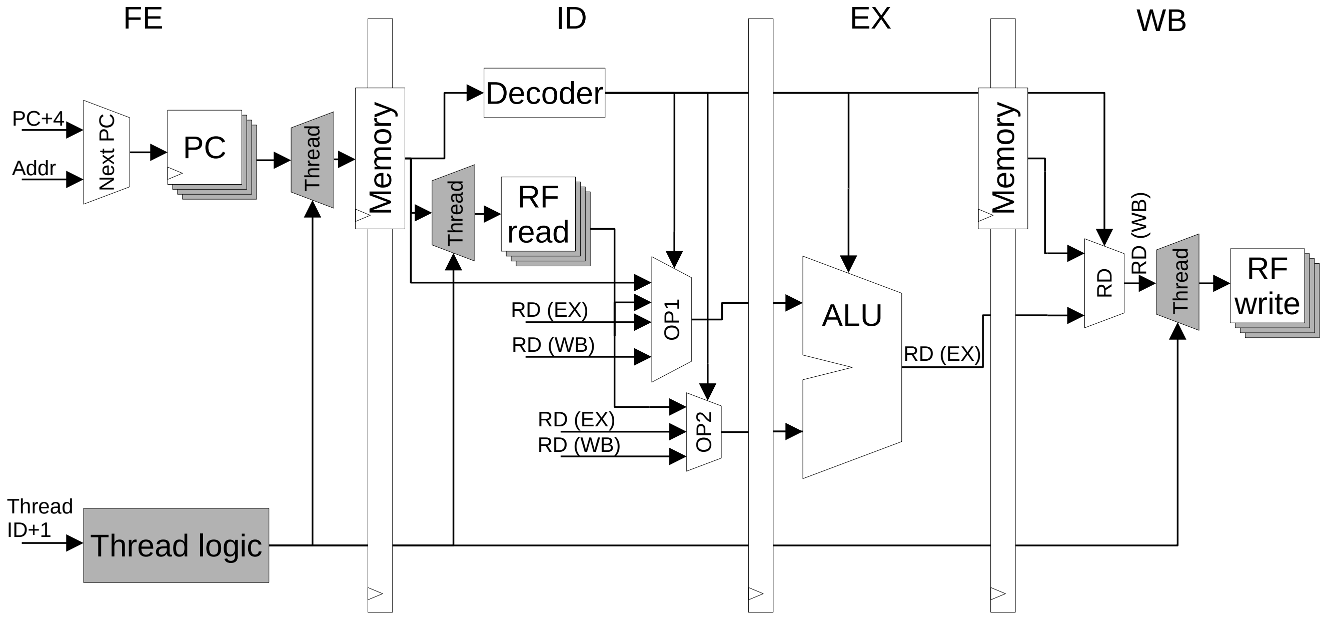


Fig. 1. Schematic of data path in modified PRET uRISC processor, grey colored components are added to the original design in order to implement multithreading

Call interrupt instruction is modified to call one of the four interrupt service routines.

Decode stage issues a call interrupt instruction if the following conditions are met: interrupt enable register is set, interrupt signal is high and both match the thread id of current instruction in decode stage. The fetched instruction of the current thread is discarded and replaced by a call interrupt, which saves the program counter of the current thread and replaces it with an address of ISR. The interrupt enable register corresponding to thread id is cleared, so no interrupt can be serviced in the thread until the ISR ends and enables interrupts by setting the interrupt enable register.

Jump interrupt instruction returns from ISR. It restores the thread to a state before the interrupt. The right program counter and interrupt enable register of the corresponding thread are set.

B. Test platform

The uRISC is a plain core. For full functionality, it is coupled with peripherals, which enables the whole platform to execute software. All peripherals are connected through the AHB3 Lite bus to the core. Any transaction on the bus takes two clock cycles, one address cycle and one data cycle. Effectively, every transaction takes only one clock cycle due to pipelining. The address is issued in execute stage and data in the writeback stage.

To fully support isolated interrupts, four programmable interrupt controllers (PIC) are present for a multithreaded platform and a single PIC for a single-threaded platform. Each PIC is coupled with a timer. The timers are original sources of interrupts on the presented platform. Both types of peripherals are connected through the AHB3 Lite interface. The peripherals support read and write access to its control registers over the bus. PICs have an additional one-bit interface for interrupt signals, which are connected directly to the core.

The memory subsystem is a crucial part of the platform. There are separate memories for the program and for data. Such a setup allows adjusting latencies for each memory independently. An approximation of complex cache memory subsystem is achieved by changing the latencies of memories which may affect the time-predictability of the platform.

TABLE I
COMPARISON OF RESOURCE UTILIZATION IN FPGA

FPGA resource type	uRISC	PRET uRISC	increase
Slice LUTs	1715	3532	206 %
Slice Registers	1416	4624	327 %
F7 Muxes	285	1117	392 %
F8 Muxes	0	512	-

TABLE II
COMPARISON OF EXECUTION TIMES ON SINGLE AND MULTITHREADED URISC

benchmark	uRISC		PRET uRISC	
	(clk)	normalised	(clk)	normalised
iir	3815	1	12052	0,79
bitcount	23161	1	73656	0,79

V. EXPERIMENTAL EVALUATION

We evaluate our modified uRISC processor in terms of increased FPGA resource allocation and in terms of real-time properties.

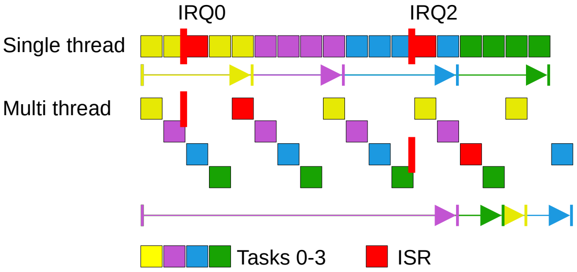


Fig. 2. Comparison of task execution on singlethread and multithreaded uRISC; Example shows task 0 (Yellow) and 2 (blue) preempted by interrupts on both platforms, completion time of task 1 (purple) and 3 (green) is prolonged by interrupts on singlethread uRISC and not influenced on multithreaded PRET-like uRISC.

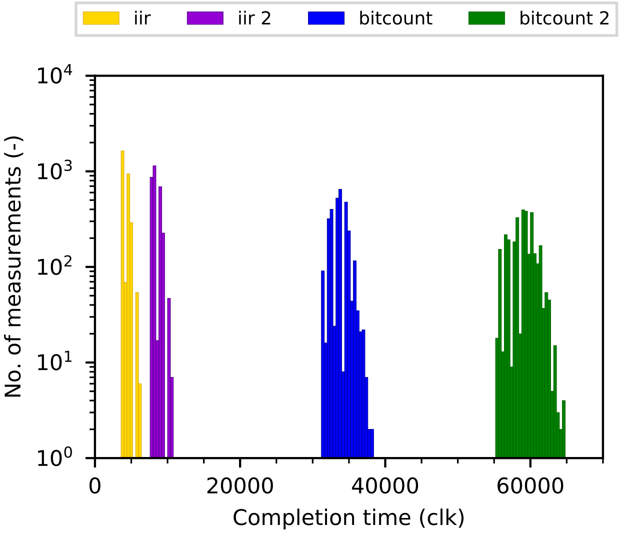


Fig. 3. Completion times of *iir* and *bitcount* benchmarks on singlethreaded core, Interrupts are enabled.

A. Resource cost

The whole platform design has been synthesized for Xilinx Artix 7 FPGA. The FPGA resource requirements loosely translate to the area. As researchers often demonstrate their designs on FPGAs, we present area requirements. Table I shows a comparison of the platforms for single and multithreaded uRISC. The multithreaded uRISC requires 206% of LUTs, 327% of registers, 392% of F7 muxes and additional 512 F8 muxes in comparison to single-threaded uRISC. If four uRISC cores were used, 400% of resources would be required, and, if it were to share the memory, memory arbitration would be required. We achieve this ratio on a simple uRISC core without branch predictor, divider or floating-point unit and essential ISA. If more complex and thus more resource-heavy processors were modified, the resource requirements ratio would be smaller. It should be noted that over 300% increase in registers is due to quadrupled register file, which could be

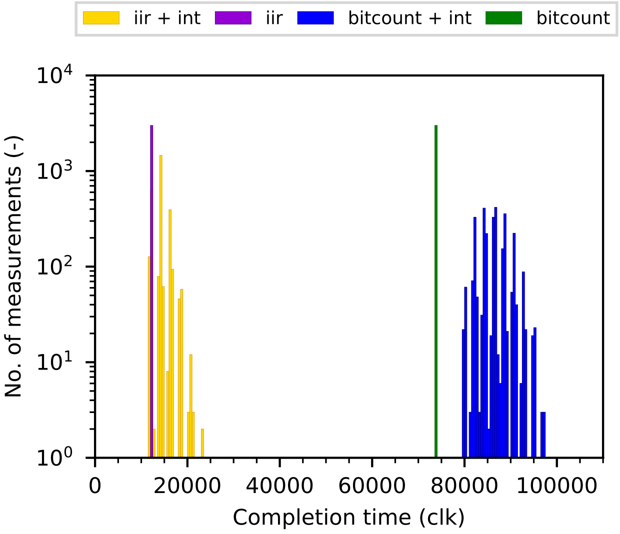


Fig. 4. Completion times of *iir* and *bitcount* benchmarks on multithreaded configuration. All four benchmarks are run simultaneously, but in a separate thread. Two threads have interrupt enabled and two disabled.

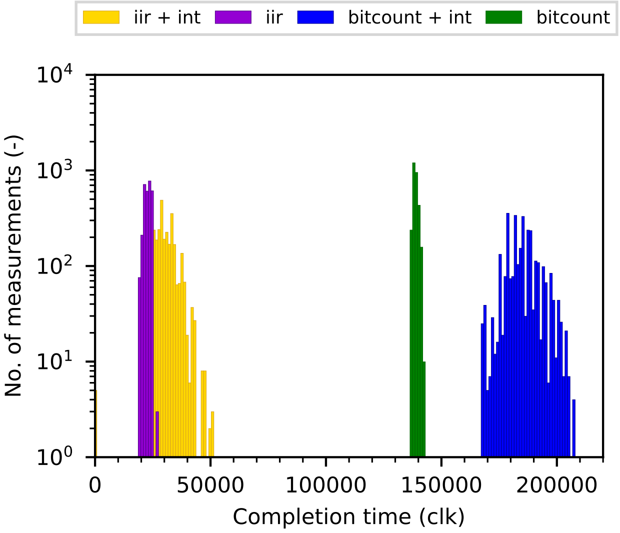


Fig. 5. Completion times of *iir* and *bitcount* benchmarks on multithreaded configuration when data memory latency is 10 clock cycles. All four benchmarks are run simultaneously, but in a separate thread. Two threads have interrupts enabled and two disabled..

mitigated by halving its size by adopting modifications similar to RISC-V extension E, which proposes to use only 16 general-purpose registers.

B. Real-time properties

We select two single-thread benchmarks from the TACleBench suite [17] to demonstrate how the real-time applications can benefit from interrupt isolation. One benchmark is *iir* and the other is *bitcount*. This selection is made to show the behaviour of tasks with short and long execution time. We do not show the rest of the benchmarks from the suite because all benchmarks are influenced by the interrupts in the same way. The benchmarks are compiled with LLVM based (Clang) Codasip compiler, which is automatically generated based on the processor description in CodAL. The compiler offers optimization presets -O0 (no optimization) through -O3 (maximal optimization). We choose to compile with optimization level set as -O2, which is commonly used by software developers and generates smaller code than -O3, which is preferred in embedded systems.

As the TACleBench does not have benchmarks, which would use interrupts, we simulate synthetic interrupts to evaluate the benefits of interrupt isolation. The interrupts are generated pseudo-randomly from the ISR. The ISR generates a pseudo-random number with uniform distribution ranging from 1 through 25 000 and sets the timer. Every benchmarking setup is run 3 000 times to get enough data to evaluate execution time jitter.

Two scenarios are benchmarked. First, for single-threaded uRISC processor, two sources of interrupts are enabled, and a simple periodic round-robin schedule of four tasks is executed, *iir*, *iir 2*, *bitcount* and *bitcount 2*. This scenario mimics a traditional approach of software threads that execute tasks non-preemptively. We measure the completion time of all tasks from the start of task 0 till the end of each task, as shown in Fig. 2. The histogram of measured execution times of this scenario is shown in Fig. 3. It is obvious that the only source of time-nondeterminism in this scenario is from ISR.

The second benchmarked scenario is a multithreaded uRISC processor with PRET-like modifications. Again four tasks are executed, but this time concurrently in four hardware threads. Two threads have interrupts enabled, and the other two execute without interrupts. In Fig. 4 we show the completion times, which directly translates to execution time for this scenario. The longer execution times caused by interrupts affect only the respective threads, and the execution of other threads remains unaffected. If we normalize the completion times of the unaffected tasks to a thread time, the execution time is shorter than on a single-threaded core, as shown in Table II. This is due to the increased efficiency of branching instructions, which no longer clear the pipeline of the consecutive instructions, as there is no dependency of consecutive instructions in the pipeline.

Next, we want to evaluate the effect of interrupt isolation under the presence of time-nondeterminism caused by increased memory access latency. Higher memory access latency is typically present in bigger CPUs. We increase the memory latency to 10 clock cycles. Therefore, each load-store instruction stalls the whole pipeline, which, with our implementation, influences all threads, not only the one accessing the memory. Fig. 5 shows the results of our benchmark in the multithreaded scenario. It can be seen that the jitter of interrupt-isolated tasks (magenta, green) is greater than zero but still significantly smaller than the jitter of interrupt-enabled tasks.

VI. Conclusion

We proposed an enhancement of an in-order single-issue processor IP by PRET-like modifications to increase time-determinism for mixed-criticality systems. We evaluated our proposal on the uRISC processor. We demonstrated the behaviour of the modified PRET uRISC on two benchmarks. We conclude that the execution time of a single task is increased proportionally to the number of hardware threads, the execution efficiency of real-time tasks is increased, and time-determinism of interrupt independent tasks can be guaranteed even under the presence of other simultaneously executing interrupt-driven tasks. The execution time jitter of interrupt independent tasks is eliminated by 100%. These results are also valid for any other single-threaded application.

The cost of our modifications lies in the increased chip/FPGA area. Specifically, when compared to the original uRISC, our implementation needs 206% of LUTs, 327% of registers, 392% of F7 muxes and additional 512 F8 muxes. This is due to multiplying the register file for storing the thread context, but those numbers are smaller than for instantiating a complete core for each thread.

Although time-determinism of the CPU is degraded when memory latencies are higher than one clock cycle, we have shown that our interrupt isolation decreases the execution time jitter even in this setting.

References

[1] T. Mitra, "INVITED time-predictable computing by design: Looking back, looking forward," in *2019 56th ACM/IEEE Design Automation Conference (DAC)*, pp. 1–4, ISSN: 0738-100X.

[2] H. Ding, Y. Liang, and T. Mitra, "WCET-centric partial instruction cache locking," in *DAC Design Automation Conference 2012*, pp. 412–420, ISSN: 0738-100X.

[3] D. Sanchez and C. Kozyrakis, "Vantage: scalable and efficient fine-grain cache partitioning," *ACM SIGARCH Computer Architecture News*, vol. 39, no. 3, pp. 57–68. doi: 10.1145/2024723.2000073

[4] G. Gracioli, R. Tabish, R. Mancuso, R. Mirosanlou, R. Pellizzoni, and M. Caccamo, "Designing mixed criticality applications on modern heterogeneous MPSoC platforms," in *31st Euromicro Conference on Real-Time Systems (ECRTS 2019)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), S. Quinton, Ed., vol. 133. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi: 10.4230/LIPIcs.ECRTS.2019.27. ISBN 978-3-95977-110-8 pp. 27:1–27:25, ISSN: 1868-8969.

[5] M. Schoeberl, B. Rouxel, and I. Puaut, "A time-predictable branch predictor," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, ser. SAC '19. Association for Computing Machinery. doi: 10.1145/3297280.3297337. ISBN 978-1-4503-5933-7 pp. 607–616.

[6] S. A. Edwards and E. A. Lee, "The case for the precision timed (PRET) machine," in *Proceedings of the 44th annual Design Automation Conference*, ser. DAC '07. Association for Computing Machinery. doi: 10.1145/1278480.1278545. ISBN 978-1-59593-627-1 pp. 264–265.

[7] M. Zimmer, D. Broman, C. Shaver, and E. A. Lee, "FlexPRET: A processor platform for mixed-criticality systems," in *2014 IEEE 19th Real-Time and Embedded Technology and Applications Symposium (RTAS)*. doi: 10.1109/RTAS.2014.6925994 pp. 101–110, ISSN: 1545-3421.

[8] M. Schoeberl, P. Schleuniger, W. Puffitsch, F. Brandner, C. W. Probst, S. Karlsson, and T. Thorn, "Towards a time-predictable dual-issue microprocessor: The patmos approach," vol. 18. doi: 10.4230/OASIcs.PPES.2011.11 p. 11.

[9] C. Sung, M. Kusano, and C. Wang, "Modular verification of interrupt-driven software," in *2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE)*. doi: 10.1109/ASE.2017.8115634 pp. 206–216.

[10] Y. Wang, L. Wang, T. Yu, J. Zhao, and X. Li, "Automatic detection and validation of race conditions in interrupt-driven embedded software," in *Proceedings of the 26th ACM SIGSOFT International Symposium on Software Testing and Analysis*, ser. ISSTA 2017. Association for Computing Machinery. doi: 10.1145/3092703.3092724. ISBN 978-1-4503-5076-1 pp. 113–124.

[11] M. Pan, S. Chen, Y. Pei, T. Zhang, and X. Li, "Easy modelling and verification of unpredictable and preemptive interrupt-driven systems," in *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*. doi: 10.1109/ICSE.2019.00037 pp. 212–222, ISSN: 1558-1225.

[12] E. Lee, J. Reineke, and M. Zimmer, "Abstract PRET machines," in *2017 IEEE Real-Time Systems Symposium (RTSS)*. doi: 10.1109/RTSS.2017.00041 pp. 1–11, ISSN: 2576-3172.

[13] Z. Prikryl, "Fast simulation of pipeline in ASIP simulators," in *2014 15th International Microprocessor Test and Verification Workshop*. doi: 10.1109/MTV.2014.18 pp. 10–15, ISSN: 2332-5674.

[14] P. Sláma, "Instruction level parallelism in modern processors," Master Thesis, BUT, 2020.

[15] M. Fajčík, "Automation of verification using artificial neural networks," Master Thesis, BUT, 2016.

[16] M. Fajcik, P. Smrz, and M. Zachariasova, "Automation of processor verification using recurrent neural networks," in *2017 18th International Workshop on Microprocessor and SOC Test and Verification (MTV)*. doi: 10.1109/MTV.2017.15 pp. 15–20, ISSN: 2332-5674.

[17] H. Falk, S. Altmeyer, P. Hellinckx, B. Lisper, W. Puffitsch, C. Rochange, M. Schoeberl, R. B. Sørensen, P. Wägemann, and S. Wegener, "TACLeBench: A benchmark collection to support worst-case execution time research," in *16th International Workshop on Worst-Case Execution Time Analysis*. doi: 10.4230/OASIcs.WCET.2016.2

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 501–510
DOI: 10.15439/2021F58
ISSN 2300-5963 ACSIS, Vol. 25

Semi-automated Algorithm for Complex Test Data Generation for Interface-based Regression Testing of Software Components

Tomas Potuzak
Department of Computer Science and Engineering/
NTIS – New Technologies for the Information Society,
European Center of Excellence, Faculty of Applied
Sciences, University of West Bohemia
Univerzitni 8, 306 14 Plzen, Czech Republic
Email: tpotuzak@kiv.zcu.cz

Richard Lipka
NTIS – New Technologies for the Information
Society, European Center of Excellence/Department
of Computer Science and Engineering, Faculty of
Applied Sciences, University of West Bohemia
Univerzitni 8, 306 14 Plzen, Czech Republic
Email: lipka@kiv.zcu.cz

Abstract—**This paper describes in detail the Complex Object Generation (COG) algorithm, which is a semi-automated algorithm for the generation of instances of classes (i.e., objects) with a complex inner structure for Java and similar languages designed for black-box testing (i.e., without available source code). The algorithm was developed and tested as a stand-alone algorithm and can be used as such (e.g., during unit testing). However, we plan to use it to generate the parameter values of generated method invocations, which is a vital part of our interface-based regression testing of software components.**

I. Introduction

SOFTWARE development utilizing components has been around about two decades. Its main idea is to construct software from isolated parts (called software components), which can interact solely using well-defined interfaces. One of the purposes is to enhance the reusability of the software parts meaning to use a software component in different applications. On the other hand, a single application often consists of components possibly from different authors [1]. Besides the benefits of software parts reuse, there are also some setbacks, especially regarding testing. Since the components in a single application can originate from different authors, testing of their correct cooperation within this application is vital [2], because such testing cannot be performed by the authors of the individual components.

The necessity for testing is even more stressed by the common situation that the individual components exist in several versions. These versions can have only subtle changes, but can also differ significantly [3]. The individual versions of the component can have different internal calculations, interact differently with other components, or can have different public interfaces. Theoretically, the changes in the internal behavior of the component should not propagate past its interface. So, there should be no influence on the working of the entire component-based application. However, the reality is different from this theory. During the development of the new version of the component, new errors can be introduced, side effects of method invocations can change, computations may become more complex (e.g., because of an improved fidelity of the results) leading to a longer computation time and a time-out expiration. Further examples could continue. For the reasons described above, a thorough regression testing should be performed whenever a new version of a component is installed to a component-based application. From this point of view, it is not important whether there are changes to the public interface of the new version of the component or not [2], [4].

In order to support this regression testing, we developed a testing approach for components without available source code (i.e., black-box testing). This is, for example, the case of third-party components, which are not open source. It should be noted that some form of source code can be obtained using the reverse engineering even when it is not at our disposal directly. However, this requires additional effort and the results may not be ideal. Even the languages such as Java, whose byte code can be transformed to source code readily, can use obfuscation techniques [5] to hamper the reverse engineering. There are also legal aspects – the reverse engineering might violate the software license.

Our approach is designed for black-box testing for the situation when an old version of a component is replaced by its new version. The aim is to determine, whether both versions exhibit the same external behavior within the component-based application of our interest [2], [3], [4]. A prototype implementation of this approach was described in [2] in detail. It is implemented in Java and designed for the OSGi [6] component model, but its core ideas can be used also for other similar component models and languages [3]. It is implemented in our Interface Analysis Tool (InAnT). The approach is based on the analysis of public interfaces of the individual software components in the component-based application. The analysis discovers all services provided by

This work was supported by University specific research project SGS-2019-018 Processing of heterogeneous data and its specialized applications.

all the components together with the methods of these services. For each method, a set of method invocations (i.e., unique combinations of parameter values) is generated. These invocations are then performed and their consequences are observed in an iterative phase. This way, the behavior of the entire application is recorded. The process is performed in the application with the old and then with a new version of the component. The comparison of the two recordings can then show any changes in the behavior of the entire component-based application (presumably caused by the installation of the new version of the component) [2], [3], [4].

For the generation of the parameter values for the method invocations, various automated approaches can be used, for example the combinatorial testing [7] or the particle swarm optimization [8]. The approach used in our prototype implementation is rather primitive [3]. For parameters of primitive data types, several common and border values are used. For general objects, only `null` value is used [2]. Even then, our approach was able to uncover changes in our two test cases [2]. Nevertheless, usage of more realistic values would significantly improve the performance of our approach [3].

It should be noted that our interface-based approach for regression testing of software components is a black-box testing approach, which means that we do not expect the access to the source codes of the software components under tests [2]. At the same time, the most problematic part is the generation of complex objects. Since there are very few existing approaches for this task, we decided to develop a semi-automated approach for it. The approach described in this paper, the Complex Object Generation (COG), explores the structure of a selected class and enables to create its new instance (object) and fill its structure using existing constructors and manual changing of the attributes via a generated graphical user interface (GUI). The generated objects can be then used as parameter values for the method invocations. The COG was implemented in InAnT. First, it was tested as a stand-alone algorithm and can be used as such (e.g., during unit testing). However, it will be incorporated into our interface-based regression testing of software components. The main idea of the COG algorithm and its initial testing were briefly described in [3]. The detailed description of the COG algorithm and all its aspects including its further testing description is the main contribution of this paper.

II. InAnT Description

As mentioned above, the interface-based regression testing of software components is designed for component-based applications and, as such, its prototype implementation is a software component itself. It was described in [2] and one of its important parts – the Deep Object Comparison (DOC) – was described in [4]. The Complex Object Generation (COG) is another important part. Nevertheless, the basic notions of the component-based software development and the basic features of the InAnT are briefly discussed in following subsections in order to make this paper self-contained.

A. Component-based Software Development

A *software component* is a black-box software entity with a well-defined public interface consisting of provided *services*. The component can but does not have to require services of other components for its functioning. The components are expected to interact using their public interfaces only. These general features are common among various types of software components. However, the specific details of the aspects, behavior, and interactions of the software components depend on the used *component model*. A specific implementation of a component model is called a *component framework*. There can be (and often are) multiple component frameworks of a single component model [1], [4].

The prototype implementation of InAnT was implemented for the Java and the OSGi component model [2], [4]. Currently, the OSGi is quite widespread in both academic and industrial fields. There are several OSGi frameworks (i.e., implementations of the OSGi model), such as Felix or Equinox [2]. The OSGi components are called *bundles*. Each bundle is a single standard `.jar` file with additional meta-information related to the functioning of the OSGi (e.g., name and version of the bundle, required and exported packages, etc.) [2], [6]. Every bundle can provide several services in the form of Java interfaces. These services with the exported packages and their content form the public interface of the bundle [2]. The OSGi is a dynamic component model – the components can be installed and uninstalled on the fly (i.e., without the necessity to restart the OSGi framework) [9]. To enable this, the OSGi framework runtime provides means [6] for the control of the lifecycle of the bundle and also for the exploration of its environment [2].

The testing of the component-based application is similar to the testing of monolithic applications with additional issues caused by the composition of the components from different authors [2]. The testing methods can be divided based on the availability of the source code for the testers [10]. Source code can be used for the preparation of the tests leading to *white-box testing*. If it is not available or not used for the test preparation, the testing is called the *black-box testing* [11]. We consider this type, since the source code can be often not available for third-party components [2].

Regardless the type of testing, its main principle is to subject the tested software component to a set of inputs, to observe the outputs, and to compare them to the expected outputs [12]. A test is described in a so-called *scenario*. The content of the scenario forms the inputs and (optionally) the expected outputs. For the black-box testing of software components, each input can correspond to an invocation of a method of a service of the tested component [2].

B. Generation of Testing Scenarios

Our interface-based regression testing is used to find any changes of the behavior of a component-based application after a new version of a software component is installed instead of its old version. The application is tested with the old version of the component and then with its new version. For both test runs, the invocations of the methods of the services of all components are generated, performed, and their consequences are recorded. The invocations and their consequences are stored to a recording – a scenario [2].

The InAnT prototype implementation has the form of a single OSGi bundle installed in the same framework as the component-based application under test. The invocation generation starts with the identification of all methods of all services of all components of the component-based application. This is achieved using the OSGi methods for the exploration of the bundles' environment and the Java reflection [2]. The found components, their services, and their methods are added into a data structure with the form of a tree [4].

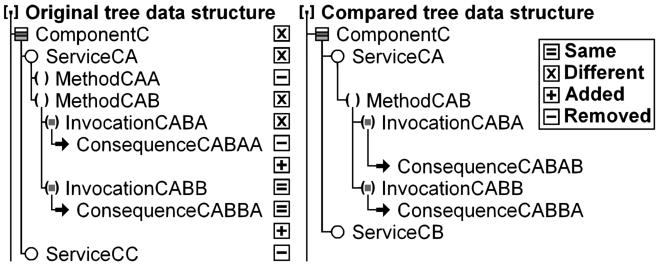
The structure is then explored and, for every method, an initial set of invocations is generated. Each invocation is represented by the values of the method parameters. In our prototype implementation, the generation of these values is very simple – several common and border values are used for the parameters of primitive data types and only the `null` value is used for the objects [2]. Each invocation is a unique combination of parameter values of a method. The generated invocations are added to the data structure, which forms the basis of the scenario [4]. The purpose of the COG algorithm described in this paper is to provide additional object values.

Once the initial invocations are inserted into the data structure, the iterative phase begins. The data structure is explored and the invocations are consecutively performed (i.e., the corresponding method is invoked with the parameter values from the invocation). For each performed invocation, its consequences are observed. There can be multiple consequences of a single method invocation. There are four observed types – a thrown exception, a return value, a subsequent invocation of another service method, and a value change in output parameters of the method [2], [4]. All the consequences, which are not yet present in the data structure, are added to the invocation, which caused them.

The subsequent invocations are also added as invocations for the corresponding method (if they are not yet present). Their parameter values come from the internal logic of the components, making them more useful than the generated parameter values. The subsequent invocation generated in the $(n-1)$th iteration is performed in the nth iteration, where it can bring new consequences, which might remain hidden if only the generated initial invocations were used [2], [4].

Nevertheless, the recording of subsequent invocations has a setback. Since the components are black boxes, the causality of the performed invocation and its subsequent invocations is not certain. For example, if there are active threads in some components, they can perform invocations of methods of other components independently on our testing, yet these invocations will be recorded. This can cause false alarms during the testing scenarios comparison (see Section II.C) [2].

The iterative phase ends when there are no new consequences added in the last completed iteration [2], [4] or the preset maximal number of iterations was performed. The filled data structure is stored as a scenario with the old version of the component to an XML file [2], [4]. The tree-like nature of the data structure can be observed in Fig. 1.



Differences on the services level (ServiceCB added, ServiceCC removed, lower levels NOT compared), on methods level (MethodCAA removed), and on consequences level (ConsequenceCABAA replaced by ConsequenceCABAB)

Fig. 1 An example two scenarios (data structures) comparison result

C. Comparison of Testing Scenarios

After the installation of a new version of a component into the application under test, the process described in Section II.B is performed again and a new scenario with the new version of the component is obtained. The scenario with the old version of the component is then loaded from the XML file and the data structures of both scenarios are compared on every level (i.e., the components level, the services level, the methods level, etc.) [2], [4].

On every level, the presence of an item in both structures is checked. If the item is present in both structures, the subtree of the item is expanded in both structures and the comparison continues in lower levels. If the item is missing in one of the data structures, this difference is reported and the lower levels are not explored further, since there is nothing to compare. The items are considered equal if and only if their subitems are equal on every sublevel [2], [4]. An example comparison result is depicted in Fig. 1.

The comparison result is also the result of the entire interface-based regression testing of software components. The differences found in the data structures indicate a change of the behavior of the component-based application under test after the installation of a new version of a component. The most important differences are on the invocations level and on the consequences level, since these differences cannot be easily detected by other means. The changes on the higher levels mean changes in the public interface of the components, which are detectable for example by an advanced static analysis (described for example in [13]) [2], [4].

III. Related Work

As it was mentioned in Section I, the most problematic part of the generation of the parameter values for the method invocations is the generation of complex objects. As far as we know, the research literature on this subject is limited, especially if it comes to the black-box testing. However, there are several works (partially) related to this subject.

A. Generation of Complex Data

There exist some tools for the complex testing input data generation. Nevertheless, they are in most cases designed for the web-based applications [14], [15], [16] and deal with data formats such as XML or JSON instead of instances. Additionally, they are not designed for black-box testing [3].

The PODAM tool [17] partially resembles the COG as it deals with standard Java objects. It enables to investigate their attributes and to fill them with random values based on their classes. The user can set the parameters of the random data generation or provide its own data where the random generation is insufficient or not desirable. However, this can be done only in the form of the user's own factory classes [17]. No GUI for direct input is provided. The tool also does not support the usage of objects created in past as attributes of the currently created objects. The source code is not required, but as the intended usage of the tool is the unit testing [17], which can be classified as white-box testing [3].

The JOP tool, developed during our past research [18], [19], enables to generate pools of complex Java objects. Its functionality is similar to the PODAM tool, but the objects created in past can be used as attributes of currently created objects [19]. The details of the object generation are described in annotations written into the source code [18], [19]. The object generation itself is possible without these annotations, but with a limited functionality. So, the knowledge of the source code is again presumed [3].

B. Exploration of Object Internal Structure

The knowledge of the objects' internal structure is necessary for their generation. This information is also vital in other fields such as memory optimization or object equality [3].

There exist papers focused on the memory optimization in programming languages with automatic memory management (e.g., Java). Their common idea is that some instances in the memory of a running program are equivalent and all equivalent instances can be replaced by a single instance without affecting the execution of the program [3], [4]. Examples can be found in [20], [21], [22], or [23]. In [20], it is pointed out that a thorough comparison of two objects (in order to determine their equality) requires checking of the graphs of the internal structures of the compared objects for isomorphism. Since this is a relatively time-consuming task and a large number of comparisons is expected for the memory optimization, many tools employ a sort of hash values or "fingerprints" of the objects to reduce the amount of necessary computations. Examples are in [20] and [22].

There are also several papers focused on object equality. In [24], the `equals()` method generator for complex objects is discussed. Deep equality is described recursively. The objects are deep equal if and only if, for all the corresponding fields of two compared objects, the deep equality holds [24]. We also employ the Deep Object Comparison (DOC) [4] for the comparison of objects in the InAnT tool. In the DOC, two objects are considered equal if and only if they have the same class, the graphs of their internal structures are isomorphic, and all values of the corresponding attributes of primitive data types in the corresponding vertices of the graphs have the same type and are equal [3], [4]. The DOC was used for the inspection of the internal structures of the instances and for their comparison during the testing described in this paper (see Section V).

C. Utilization of GUI for Setting Attribute Values

The employing a generated GUI for the data objects attributes values is discussed in several papers as well.

In [25], focused on a black-box testing of software components in .NET, the generation of the GUI for each component is discussed. This GUI enables an easier setting of input values and overall testing of the components. Reflection is used for the finding of the classes, methods and input/output parameters of the components [25].

With the FXForm2 tool, it is possible to automatically generate JavaFX forms from Java beans and to link the created GUI form fields with the Java bean properties [26].

IV. Description of Complex Objects Generation

The complex object generation (COG) is a semi-automated algorithm for generation of a new instance of a selected class, which source code is not known. For this purpose, an existing constructor is invoked with the parameters defined by the user with a generated GUI form. Then, the attributes of the created instance can be changed by the user with another generated GUI form. The algorithm is recursive, so the user can set both primitive and reference (object) values [3].

In comparison to more automatic tools described in Section III.A, the significant involvement of the user in the COG algorithm may seem like a setback. However, during the black-box testing, for which the COG algorithm is intended, the expertise of the user can be vital for the creation of instances with (at least partially) realistic attribute values. The user can utilize information, such as specification, vague textual description, or documentation, which may be at his or her disposal, but is not extractable automatically. He or she can also better interpret hints such as names of the attributes. Although the user can (unwittingly) introduce errors into generated objects, we consider the involvement of the user in the very generation of the objects an advantage.

The COG algorithm is designed for the generation of complex object parameter values for method invocations in our interface-based regression testing of software components. Nevertheless, it was first implemented and tested as a stand-alone algorithm usable wherever complex objects as input data are needed (e.g., during unit testing) [3].

A. Description of Data Structure

For the generation of a new instance of the input class, the COG algorithm requires a data structure ensuring the functioning of the algorithm and also enabling the storing of the instance generation process to disk. From the stored info-

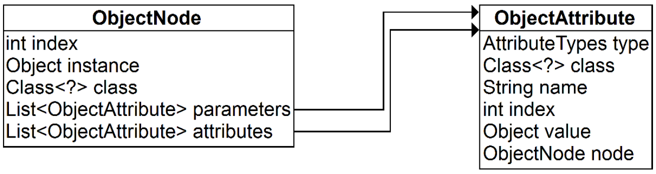


Fig. 2 The data structure for the storing of the generated instance

rmation, it is possible to reconstruct the generated instance in memory (see Section IV.G). The data structure consists of instances of two classes – one for the storing of the generated instance (called `ObjectNode`) and the second for the storing of the instance attribute values (called `ObjectAttribute`) [3]. Their attributes are depicted in Fig. 2.

The `ObjectNode` stores the information about the class of the generated instance, the list of the parameters of the constructor, which were used for the creation of the generated instance, the list of the attributes of the generated instance (non-static only), and the generated instance itself. The last attribute is the index in the *all nodes list*. In this list, all the generated `ObjectNode` instances are stored. The index also plays a role during the storing and loading to/from the disk (see Section IV.G for details). So, an `ObjectNode` instance contains all the information necessary for the creation of the generated instance [3].

The elements of the list of constructor parameters and of the list of attributes are the instances of the `ObjectAttribute` class. Each instance contains the information about the data type (i.e., the class) of the field (i.e., an instance attribute or a constructor parameter), the type of the field value (set internally or set manually), and the name and the index of the field. The name is used as a unique identification of the instance attributes and the index is used as a unique identification of the array cells (see Section IV.C) and the constructor parameters. Each `ObjectAttribute` instance also contains the value of the field, which can be a primitive value or a reference to an object. The last attribute is the reference to an instance of the `ObjectNode` class. This reference is set to `null` if the attribute is of primitive data type or its value was not specified by the user (i.e., it was set internally by the constructor). Otherwise, this last attribute points to the instance of the `ObjectNode` class describing the creation of the corresponding objects [3].

B. Description of Algorithm

The current version of the COG algorithm is implemented in Java and uses Java reflection [27] for the exploration of the classes' contents [3]. The reflection extracts the information from the bytecode, no source code is necessary. Hence, it is perfectly suited for our situation when the source code is not available. A consequence is that the COG algorithm is described with the limitations of the Java reflection in mind. For similar languages with available reflection (such as .NET platform), small changes in the implementation of the algorithm would be probably necessary, but its general idea should be utilizable in these languages as well [3].

```
createInstance(class, genericTypes, allNodes) {
  if (class.isInterface()) {
    classes = findClassesFor(class)
    if (classes.isEmpty()) {
      break; //Cannot continue without class
    }
    selected = readFromInput() //User input
    class = classes[selected]
  }
  node = createNode(class, genericTypes)
  if (!class.isArray()) {
    node.setArray(false)
    constructors = node.listConstructors()
    selected = readFromInput()
    constructor = constructors[selected]
    parameters = constructor.listParameters()
    processFields(parameters, allNodes)
    node.setConstructorParameters(parameters)
    node.createInstance(constructor, parameters)
    attributes = node.listAttributes()
    processFields(attributes, allNodes)
    node.setAttributes(attributes)
  }
  else {
    node.setArray(true)
    arrayLength = readFromInput()
    node.createArray(arrayLength)
    indices = node.listIndices()
    processFields(indices, allNodes)
    node.setAttributes(indices)
  }
  allNodes.add(node)
  return node
}
```

Fig. 3 Pseudocode for the main COG algorithm

Assume now that we want to generate parameter values for a method invocation and one of the parameter types is a class. The COG algorithm enables to create an instance of this class and to store it in the form of an instance of the `ObjectNode` to the all nodes list. The elements of this list (more specifically, the contained generated instances) can be used as constructor parameter values or attribute values of other generated instances. Before the generation of the first instance, the all nodes list is empty. The input of the COG algorithm is the class, for which the instance shall be generated, and the information about the generic type(s) if the class is parameterized (see Section IV.D) [3]. The main algorithm is depicted in Fig. 3. There are two main steps.

In the first step, the `ObjectNode` instance is created and all constructors available for the class being generated are found using the reflection. The constructors vary in the count and/or types of their parameters [3]. The names of the parameters are not stored in the bytecode, so they cannot be determined using the reflection. The list of the constructors (each constructor represented by its parameter types) is displayed to the user and he or she selects one of them.

The user then sets the values of the parameters of the selected constructor in a GUI form. For the parameters of primitive data types, he or she inputs the values directly or can use the default value (zero). For the object parameters, the user can select an instance of the `ObjectNode` from the all nodes list if there are any applicable instances. The applicable instances of the `ObjectNode` class must contain a generated instance compatible with the data type of the constructor parameter. Only the applicable instances are displayed to the user. If there are no applicable instances in the all nodes list (besides the default `null` value) or there are some, but the user does not wish to use any of them, he or she can create a new instance for the constructor parameter recursively using the COG algorithm.

Once all the constructor parameters are inputted, they are stored in the list of the constructor parameters in the `ObjectNode` instance, each parameter as an `ObjectAttribute` instance. Since the names of the parameters are not known (see above), the parameters are identified by their order (stored as an index). The constructor is then invoked using the reflection and the new instance is created. This instance is then stored to the `ObjectNode` instance.

In the second step, all the non-static attributes of the instance being generated are found using the reflection. The information about them including their names and values are stored as the instances of the `ObjectAttribute` class to the list of attributes of the `ObjectNode` instance. The list of the attributes is displayed to the user as a GUI form. The user can change the values of the attributes similarly to the values of the constructor parameters (see above). So, the attributes of primitive data types can be set directly and the object parameters can be selected from the all nodes list or created recursively using the COG algorithm. The main difference is that the values (both primitive and object) can already have meaningful values set by the invoked constructor. So, the user can change only some values or no values at all [3]. In the latter case, the instance being generated is created solely using its constructor. The processing of constructor parameters and generated instance attributes is depicted in Fig. 4.

```
processFields(fields, allNodes) {
  for (field: fields) {
    if (field.isPrimitive() || field.isEnum()) {
      value = readFromInput() //User input
      field.setValue(value)
    }
    else {
      if (field.isInterface()) {
        classes = findClassesFor(field)
        selected = readFromInput()
        field = classes[selected]
      }
      createNew = readFromInput()
      if (createNew) {
        value = createInstance(field.class,
          field.genericTypes, allNodes)
        field.setValue(value)
      }
      else {
        values = allNodes.findValuesFor(field)
        selected = readFromInput()
        field.setValue(values[selected])
      }
    }
  }
}
```

Fig. 4 Pseudocode of the parameters and attributes processing

```java
import java.awt.Point;

public interface IShape {
  public Point getPosition();
}

public class Circle implements IShape {
  private int radius;
  private Point position;

  public Circle(int radius, Point position) {
    this.radius = radius;
    this.position = position;
  }

  @Override
  public Point getPosition() {
    return position;
  }

  @Override
  public String toString() {
    return "Circle at " + position + ", r = " +
      radius;
  }
}
```

Fig. 5 The Java codes of the `IShape` interface and the `Circle` class

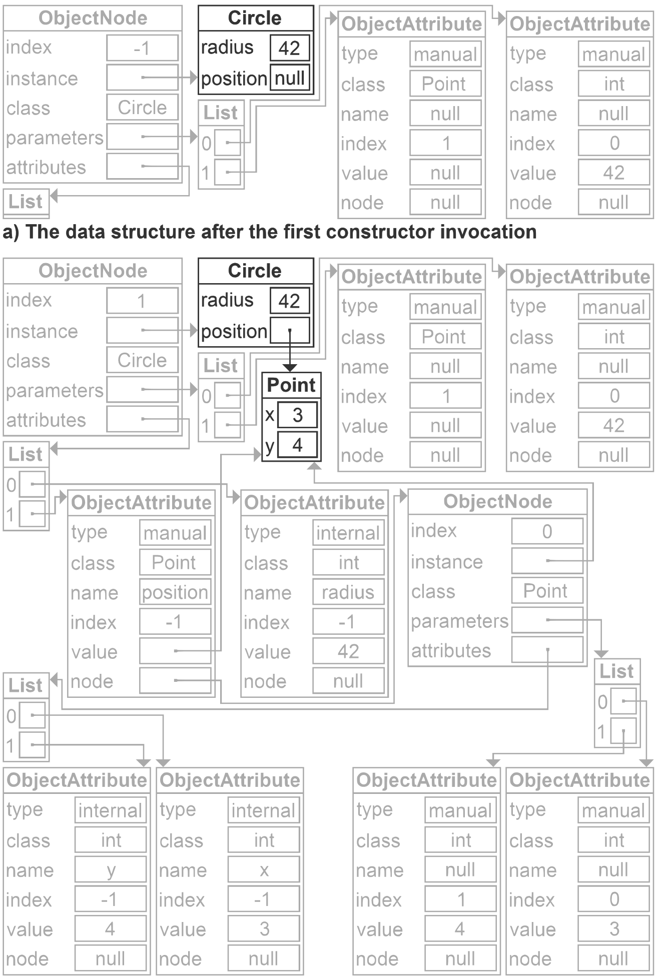
When the user changes the value of an object attribute, the reference to the corresponding `ObjectNode` instance (containing the new value) is set. However, when the user does not change the value of an object attribute, the reference to the `ObjectNode` remains set to `null`. The reason is that the value of this object attribute was created outside of the control of the COG algorithm. For example, it is not known, which constructor was used for the object creation and which parameter values were used. Any changes to the attribute values are stored to the list of the attributes in the `ObjectNode` instance and also directly to the generated instance.

At this point, the `ObjectNode` instance is added to the all nodes list. The finished instance referred from this `ObjectNode` instance can be used as a parameter value for the method invocation.

A simple example of the creating an instance of the `Circle` class (see Fig. 5) using the COG algorithm is depicted in Fig. 6. The black color is used for the generated instances and the gray color for the COG data structures.

We assume that the all nodes list is empty and the user chooses the first (and only) constructor. The `ObjectNode` instance for the `Circle` instance is created and the user must specify the constructor parameters. The `radius` parameter is of primitive data type and is set directly to 42. However, the `position` parameter is an object. Since the all nodes list is empty at this point, the user can select `null` value or can create a new `Point` instance using the COG algorithm. He or she chooses the `null` value. A new `Circle` instance is created and stored together with the constructor parameters to the `ObjectNode` instance (see Fig. 6a).

The user then inspects the attribute values of the `Circle` instance and sets a new value to the `position` attribute. He or she creates a new `Point` instance using the COG algori-



a) The data structure after the first constructor invocation

b) The data structure at the COG end

Fig. 6 The changes to the data structures

thm. The user chooses the constructor of the `Point` class with two `int` parameters. The `ObjectNode` instance for the `Point` instance is created and the user sets constructor parameters to 3 and 4. A new `Point` instance is created and stored with its constructor parameters to the `ObjectNode` instance. The user does not perform any changes of the attribute values and they are only stored to the `ObjectNode` instance. The `ObjectNode` instance for the `Point` instance is added to the all nodes list with index set to 0. The COG algorithm for the `Point` class ends and the created instance is set as the new value of the `position` attribute of the `Circle` instance. The user does no further changes. The attributes are stored to the `ObjectNode` instance for the `Circle` instance, which is then added to the all nodes list with index set to 1 (see Fig. 6b). It can be observed that the `ObjectNode` instance is created only for the attributes set by the user. The remaining attributes are considered values only, regardless of their data type (primitive or object).

The description of the COG algorithm and the example above shows only the most straightforward course of the COG algorithm. However, there are several aspects, which should be mentioned and which are discussed in following subsections. Some can be seen in the pseudocode in Fig. 3 and 4.

C. Array Handling

One of the important aspects is that the main input of the COG algorithm does not have to be only a class. It could be also an interface, which is discussed in Section IV.E, or an array. In Java, the arrays are basically instances of special classes containing predefined attributes (such as length of the array) and the indexed elements.

The COG algorithm handles the arrays (regardless whether of primitive or object data type) as classes with two main differences. The arrays do not have constructors, which can be used for the creation of a new instance of the array. However, it is possible to create an array of a specified component type with a specified length using the reflection. Hence, the user only specifies the length of the array instead of choosing the constructor. The elements of the array are not stored as named attributes, but instead as indexed cells. However, these cells are stored as the `ObjectAttribute` instances (one per array cell), only the indices are stored as the identification instead of the nonexistent attribute names. Besides these two differences, the COG algorithm proceeds in the same manner for the arrays as for the classes.

Since the multidimensional arrays are "arrays of arrays" in Java, they are handled in the same manner as the one-dimensional arrays, but with arrays in the cells.

D. Generics Handling

The class, which is the main input of the COG algorithm, can be generic, meaning that it has one or more parameter types. These parameter types cannot be determined universally using the Java reflection, but can be extracted in several cases. More specifically, it is possible to determine the parameter types for the parameters of the constructors and methods, for the return values of the methods, and for the attributes of a class. Since these cases cover all needs of the COG algorithm, we employ the parameter types checking to allow the user to select only compatible instances of generic classes.

The parameter types of input class are passed as a separate parameter of the COG method, since this information is not stored in the class. For this reason, it would not be possible to use the generic parameters for any general class with unknown parameter types. However, since the COG algorithm is designed for the generation of instances for method parameters, the parameter types of the class can be determined from the method parameter using the reflection.

E. Interface-Implementing Class Searching

Another quite a common possibility is that the method parameter, for which the COG algorithm shall generate the instance, is an interface, not a class. In that case, it is not possible to create the instance of this interface and it is not known, which class implementing this interface shall be used for the instance creation instead. The reason is that the input of the COG algorithm is the class, not its instance. The problem can arise also for the parameters of the constructor or during the changing of the value of an attribute.

```
findClassesFor(interface) {
  classes = findClassesInDirectories()
  classes.addAll(findClassesInJars())
  classesForInterface = List()
  for (c: classes) {
    if (interface.isAssignableFrom(c) &&
        !c.isInterface()) {
      classesForInterface.add(c)
    }
  }
  return classesForInterface
}
```

Fig. 7 Pseudocode of the finding the classes implementing an interface

For this situation, it is first checked, whether the input class is an interface. If so, the available classes implementing this interface are found and the user can select one of these classes. The selected class is then used instead of the interface in the remaining course of the COG algorithm (see Fig. 3). If there are no classes implementing the interface, the algorithm ends with a failure.

In the current implementation (see Fig. 7), all directories and `.jar` files in manually specified paths are searched for the `.class` files and the availability of each corresponding class is checked by the class loader. So, it is possible that some classes implementing the interface are missed. A straightforward solution is to find all classes implementing the interface available in the application context. The problem is that the function for the discovering of all implementing classes of an interface is not directly available in Java reflection. However, since the COG algorithm is primarily intended for the OSGi, it may be possible to use its services and find all implementing classes in all bundles of the OSGi framework. If we want to achieve similar task in plain Java, it is possible using the exploration of the classpath or using third-party solutions, such as [28].

A last resort solution is to create a mockup implementation of the interface using for example the `Proxy` class from the Java reflection [29]. This can be done manually for each interface, which can be very time consuming. Another possibility is to create a generic implementation utilizable for all interfaces. In both cases, it cannot be expected that the mockup implementation of the interface will have similar behavior to a real implementation. To find and implement the optimal solution is a part of our future work.

F. Exception Handling

In each phase of the COG algorithm, the user can use a `null` value instead of the creation or the selection of an instance. This value may be valid in many cases, but can also cause problems, usually in the form of a thrown exception. The problems occurring while utilizing the generated instance for the testing is outside the scope of this paper, but the problems can occur also during its generation – when the `null` value is used as a constructor parameter and the constructor does not permit this value. The problem does not occur while setting an instance attribute to the `null` value, since no methods of the generated instance are invoked.

The problem can occur not only because of the `null` value. Other values, such as a primitive type value outside an acceptable range or an object with unexpected attribute values can cause similar problems, typically in form of a thrown exception. For this reason, when an exception is thrown during the COG algorithm, the user is notified and can repeat the action, which caused the exception (typically invocation of a constructor) with different parameters. The user can also choose to interrupt the COG algorithm.

G. Storing and Loading to/from Disk

Since the creating of a very complex object can be quite time consuming for the user, it is possible to save the created `ObjectNode` instances from the all nodes list to disk. The `ObjectNode` instances are stored to an XML file, which is legible by humans. The generated instances contained in these nodes are not stored. Similarly, the values of their attributes not changed by the user are not stored. The storing of this information would require full scale serialization of general objects.

More importantly, storing this information is not necessary, because the generated instances can be reconstructed in the memory during the loading of the XML file using the remaining attribute values of the `ObjectNode` instances. These values are the parameter values of the utilized constructor and the attribute values changed by the user. All these values are either stored as other `ObjectNode` instances or are of primitive data type meaning they can be easily stored in a textual form. The references to the `ObjectNode` instances are replaced by the indices in the XML file. These indices correspond to the order of the instances in the all nodes list (and are also stored in each instance).

When the `ObjectNode` instances are loaded from the file, they are created in the order they were in the all nodes list prior to their storing. For each `ObjectNode` instance, the contained generated instance is created using the constructor corresponding to the stored parameter types and their stored values. At this point, the attribute values not changed by the user should be equal to their values prior to the saving to the XML file. Then, the attribute values changed by the user are set directly to the values stored in the XML file for the primitive data types and set to the correct references for the objects.

V. Tests and Results

The functioning of the COG algorithm was tested using two sets of tests. In first set of tests, the very functioning of the algorithm was demonstrated using several different classes and arrays. In second set of tests, the storing and the loading of the generated instances were investigated. All the tests were performed on a notebook with dual-core Intel i5-6200U at 2.30 GHz with 8 GB of RAM, and a 250 GB SSD and 500 GB HDD. The installed software was Windows 7 SP1 64bit, Java 1.8 (64 bit), and Equinox OSGi framework.

A. Object Generation Testing

The correct functionality of the COG algorithm was tested in two scenarios. In the first scenario, the Circle class (see Fig. 5) and the Rectangle class, which also implements the IShape interface (see Fig. 5), together with the standard Point and ArrayList classes from the Java Core API were used. All the objects were created together in a single run, so all were placed to the all nodes list.

The user used the COG algorithm to create the instances in several steps (<X> denotes a reference and index):

1. Create <0> = Point(10, 20)
2. Create <1> = Point(<0>)
   a. Set x = 42
3. Create <2> = Circle(30, <0>)
4. Create <4> = Shape() (interface)
   a. Select Rectangle as implementing class
   b. Create <3> = Rectangle(3, 4, null)
      i. Set sideA = 10
      ii. Set position = <1>
5. Create <5> = ArrayList()
   a. Create <4> = Object[2]
      i. Set 0 = <0>
      ii. Set 1 = <3>
   b. Set elementData = <4>
   c. Set size = 2 *(c. Set size = 3)*

The structure of the instances, which was intended to be generated, is shown in Fig. 8. The numbers of the instances are the indices in the all nodes list of the ObjectNode instances, in which the generated instances are contained. To determine, whether the created structure is correct, the entire ArrayList was printed using its toString() method. Its result is depicted in Fig. 9. The resulting structure was also inspected using the DOC algorithm [4]. The actual and the expected structures were manually compared. The structures were identical.

In order to show the possible issues caused by the direct involvement of the user, the instances were created again using the steps above, but with the last step (5.c) displayed using italics in parentheses. That means that the size of the ArrayList was set incorrectly (not corresponding to the actual size of its inner array). This inconsistency leads to an exception (a ConcurrentModificationException) when the toString() method is invoked. This shows that the user an easily set the attributes of the instances inconsistently. Since the COG algorithm does not understand the internal functioning of the created objects, it is not possible to perform an automated consistency control. So, the user must proceed with caution.

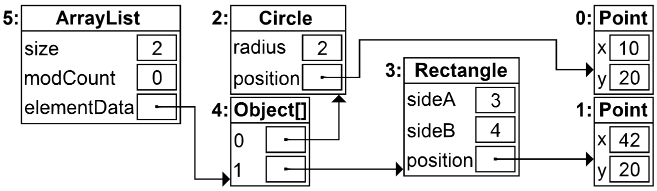


Fig. 8 The expected structure of the generated instances

```
[Circle at java.awt.Point[x=10,y=20], r = 30,
Rectangle at java.awt.Point[x=42,y=20]),
a = 3, b = 4]
```

Fig. 9 Result of the toString() method of the created ArrayList

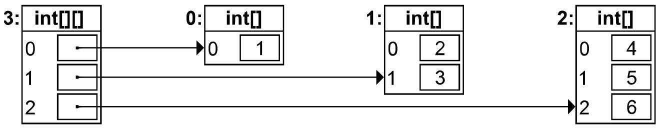


Fig. 10 The expected structure of the 2D array

```
[[1], [2, 3], [4, 5, 6]]
```

Fig, 11 Result of the Arrays.deepToString() method

In second scenario, the user attempted to create 2D array with three rows and the length of each row increasing with its index (see Fig. 10). The following steps were taken:

1. Create <3> = int[3][]
   a. Create <0> = int[1]
      i. Set 0 = 1
   b. Set 0 = <0>
   c. Create <1> = int[2]
      i. Set 0 = 2
      ii. Set 1 = 3
   d. Set 1 = <1>
   e. Create <2> = int[3]
      i. Set 0 = 4
      ii. Set 1 = 5
      iii. Set 2 = 6
   f. Set 2 = <2>

The expected structure of the 2D array is depicted in Fig. 10. To verify the correctness of the created array, it was printed using Arrays.deepToString() method. The result is depicted in Fig. 11. Again, the resulting structure was also inspected using the DOC algorithm [4] and manually compared to the expected structure. The structures were again identical.

B. Storing and Loading Testing

To tests the correct functionality of storing and loading to/from the XML file, the ObjectNode instances stored in the all nodes list in first scenario (see Section V.A) were saved to an XML file and then loaded from it. The original all nodes list was copied elsewhere prior the XML loading to preserve the original generated instances. These original generated instances were compared to the saved and loaded

TABLE I THE RESULT OF THE COMPARISON OF THE ORIGINAL GENERATED INSTANCES AND THE SAVED AND LOADED GENERATED INSTANCES

Instance index	Instance class	DOC result
0	Point	equal
1	Point	equal
2	Circle	equal
3	Rectangle	equal
4	Object[]	equal
5	ArrayList	equal

generated instances using the DOC algorithm (each original generated instance compared to its corresponding saved and loaded counterpart). The results of the comparisons are summarized in Table I. It can be observed that all the pairs of the corresponding generated instances are equal suggesting that the storing and loading works well.

VI. Conclusion and Future Work

In this paper, we described the COG algorithm for the generation of complex objects utilizable as parameter values for the generated method invocations. The algorithm is semi-automated and the user actions are required. However, since the approach is intended for the black box testing, the expertise of the user can help to create instances with (at least partially) realistic attribute values. The generated instances can be stored to the disk for the future utilization.

In our future work, we plan to improve the handling the cases when the input class for the algorithm is in fact an interface. We also plan to enhance the user comfort by storing the information about failures. For example, when a value inputted by the user leads to an exception, this information is stored and used for a warning when the user attempts to make the same mistake again. We will also incorporate the COG algorithm to our interface-based regression testing of software components.

References

[1] C. Szyperski, D. Gruntz, and S. Murer, Component Software – Beyond Object-Oriented Programming, ACM Press, New York, 2000.

[2] T. Potuzak, R. Lipka, and P. Brada, "Interface-based Semi-automated Testing of Software Components," in Proceedings of the 2017 Federated Conference on Computer Science and Information Systems, Prague, September 2017, pp. 1335-1344, http://dx.doi.org/10.15439/-2017F139

[3] T. Potuzak and R. Lipka, "Algorithm for Generation of Complex Test Data for Interface-based Regression Testing of Software Components," SAC '21: Proceedings of the 36th Annual ACM Symposium on Applied Computing, Virtual Event, Republic of Korea, March 2021, pp. 1305-1308, http://dx.doi.org/10.1145/3412841.3442118

[4] T. Potuzak and R. Lipka: "Deep Object Comparison for Interface-based Regression Testing of Software Components," in Proceedings of the 2018 Federated Conference on Computer Science and Information Systems, Poznan, September 2018, pp. 1053-1062, http://dx.doi.org/10.15439/2018F51

[5] J. T. Chan and W. Yang, "Advanced obfuscation techniques for Java bytecode," Journal of Systems and Software, vol. 71, No. 1-2, 2004, pp. 1-10, http://dx.doi.org/10.1016/S0164-1212(02)00066-3

[6] The OSGi Alliance, OSGi Service Platform Core Specification, release 4, version 4.2, 2009.

[7] M. Bures and B. S. Ahmed, "On the effectiveness of combinatorial interaction testing: A case study," in 2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), July 2017, pp. 69–76, http://dx.doi.org/10.1109/QRSC.-2017.20

[8] B. S. Ahmed, L. M. Gambardella, W. Afzal, and K. Z. Zamli, "Handling constraints in combinatorial interaction testing in the presence of multi objective particle swarm and multithreading," Information and Software Technology, vol. 86, pp. 20–36, 2017, http://dx.doi.org/10.1016/j.infsof.2017.02.004

[9] D. Rubio, Pro Spring Dynamic Modules for OSGi™ Service Platform, Apress, USA, 2009.

[10] G. J. Myers, T. Badgett, and C. Sandler, The Art o Software Testing, Third Edition, John Wiley and Sons, Inc., Hoboken, 2012.

[11] P. G. Sapna and H. Mohanty, "Automated Scenario Generation based on UML Activity Diagrams," International Conference on Information Technology, 2008, December 2008, pp. 209–214, http://dx.doi.org/10.1109/ICIT.2008.52

[12] S. J. Cunning and J. W. Rozenbiit, "Test Scenario Generation from a Structured Requirements Specification," IEEE Conference and Workshop on Engineering of Computer-Based Systems, 1999, Proceedings, March 1999, pp. 166–172, http://dx.doi.org/10.1109/-ECBS.1999.755876

[13] K. Jezek, L. Holy, A. Slezacek, and P. Brada, "Software Components Compatibility Verification Based on Static Byte-Code Analysis," 39th Euromicro Conference Series on Software Engineering and Advanced Applications, Santander, September 2013, pp. 145-152, http://dx.doi.org/10.1109/SEAA.2013.58

[14] Mockaroo. Accessed: 2018-35-05. [Online]. Available: https://www.mockaroo.com

[15] Dtm test xml generator. Accessed: 2018-05-05. [Online]. Available: http://www.sqledit.com/xmlgenerator

[16] Redgate. Accessed: 2018-03-05. [Online]. Available: http://www.red-gate.com/products/sql-development/sql-data-generator

[17] Podam - pojo data mocker. Accessed: 2018-03-05. [Online]. Available: https://github.com/mtedone/podam

[18] R. Lipka, "Automated Generator for Complex and Realistic Test Data," in 2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), July 2017, pp. 628-629, http://dx.doi.org/10.1109/QRS-C.2017.122

[19] R. Lipka and T. Potuzak, "Automated generator for complex and realistic test data - a case study," in Communication Papers of the 2018 Federated Conference on Computer Science and Information Systems, Poznan, September 2018, pp. 1053-1062, http://dx.doi.org/-10.15439/2018F214

[20] D. Marinov and R. O'Callahan, "Object Equality Profiling," in Proceedings of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications, Anaheim, October 2003, pp. 313-325, http://dx.doi.org/10.1145/-949305.949333

[21] A. Infante and A. Bergel, "Object Equivalence: Revisiting Object Equality Profiling (An Experience Report)," in Proceedings of the 13th ACM SIGPLAN International Symposium on Dynamic Languages, Vancouver, October 2017, pp. 27-38, http://dx.doi.org/-10.1145/3170472.3133844

[22] G. M. Rama and R. Komondoor, "A Dynamic Analysis to Support Object-Sharing Code Refactorings," in Proceedings of the 29th ACM/IEEE international conference on Automated software engineering, Vasteras, September 2014, pp. 713-723, http://dx.doi.org/10.1145/2642937.2642992

[23] M. J. Steindorfer and J. J. Vinju, "Performance Modeling of Maximal Sharing," in Proceedings of the 7th ACM/SPEC on International Conference on Performance Engineering, Delft, March 2016, http://dx.doi.org/10.1145/2851553.2851566

[24] N. Grech, J. Rathke, and B. Fischer, "JEqualityGen: Generating Equality and Hashing Methods," in Proceedings of the ninth international conference on Generative programming and component engineering, Eindhoven, October 2010, pp. 177-186, http://dx.doi.org/10.1145/1942788.1868320

[25] F. Naseer, S. U. Rehman, and K. Hussain, "Using Meta-data Technique for Component Based Black Box Testing," in 2010 6th International Conference on Emerging Technologies, Islamabad, 2010, pp. 276–281, http://dx.doi.org/10.1109/ICET.2010.5638474

[26] Dynamic JavaFX form generation. Accessed 2019-05-02. [Online]. Available: https://github.com/dooApp/FXForm2

[27] I. R. Forman, N. Forman, Java Reflection in Action, Manning Publications, 2004.

[28] Java runtime metadata analysis. Accessed 2019-05-03. [Online]. Available: https://github.com/ronmamo/reflections

[29] Class Proxy. Accessed 2019-05-03. [Online]. Available: https://docs.oracle.com/javase/8/docs/api/java/lang/reflect/Proxy.html

DOI: 10.15439/2021F106
ISSN 2300-5963 ACSIS, Vol. 25

A Time-Sensitive Model for Data Tampering Detection for the Advanced Metering Infrastructure

José Miguel Blanco, Bruno Rossi, and Tomáš Pitner
Department of Computer Systems and Communications
Faculty of Informatics, Masaryk University
Brno, Czech Republic
{jmblanco, brossi}@mail.muni.cz, tomp@fi.muni.cz

***Abstract*—Smart Grids offer multiple benefits: efficient energy provision, quicker recoveries from failures, etc. Nevertheless, there is risk of data tampering, unsolicited modification of the data of the smart meters. The main aim of this paper is to provide a model for processing the smart meter data that flags any energy consumption level that could be indication of data tampering. The proposed model is time-sensitive, allowing for tracking the energy usage along time, thus making possible the detection of long-lasting abnormal levels of energy consumption. Such model can be integrated in an anomaly detection system and in a semantic web reasoner.**

I. Introduction

SMART Grids (SGs) are modern power grids based on the integration of cyber and physical systems that enable efficient transmission of electricity, constant monitoring and self-healing properties in case of failures, with the overall aim to provide smart services and reduced costs for utilities and consumers [1], [2].

From the side of the connection between consumers and service operators, an important part of SGs is the Advanced Metering Infrastructure (AMI) that is constituted by smart meters and the communication infrastructure for dealing with bi-directional communication between smart meters, service operators and energy consumers/prosumers. Smart meters became over time a central point for the provision of smart services to energy consumers. However, the wide diffusion has also increased several concerns for service operators, such as the needs of securing the devices, dealing with privacy concerns about data usage, and avoiding potential risks of energy theft.

In this paper, we deal with potential compromission of smart meters with the purpose of altering the power consumption readings in so-called data tampering activities with the aim to gain some benefits or to harm the overall network stability by means of data injection attack [3]. Attackers can either compromise the hardware devices locally, injecting false data packets sent to control centers or modify data exchanged in other parts of the SGs infrastructure in so-called data injection attacks [3].

The proposed model is intended to be used as the basis for the implementation of algorithms to prevent data tampering from the side of energy service providers. The main characteristics that the model offers are twofold. Firstly, it can model a minimum and maximum energy consumption thanks to the modal operators. This allows to flag any energy usage that might be too big or too small and set up an alarm. Furthermore, the model also is able to track statements regarding energy usage along the time, allowing for the implementation of time-sensitive algorithms. This aspect increases the probability of detecting a real case of data tampering, as any peak or valley in the consumption would not be enough to set off an alarm. Peaks and valleys in energy usage are to be expected, but not when they last for a long time. Finally, it is important to note that the model has been implemented from a perspective of converting the data generated by the nodes into the semantic web. This means that one could be able to use the data generated by those devices, processed by the model, and input it into a semantic web reasoner, allowing for further automation and also a much better and extensive usage of the naturally generated data.

We have the following main contributions in this paper:

- Definition of a formal model based on temporal logic for data tampering of smart meters data;
- Theoretical and practical proofs of concept of the model based on sample data from UMass Smart* Dataset [4];

The paper is structured as follows. In Section II we define the concept of SGs, the importance of smart meters and the concept of data tampering for either energy theft or for reasons of false data injection attacks. In Section III we go through several related works of modelling/detecting data tampering for smart meters. In Section IV we define the temporal logic model for data tampering for smart meters, while in section V we give both a theoretical proof and a practical one based on the publicly available datasets of power consumption data. In Section VI we provide the discussion about the formal results of the model. In Section VII we discuss the impact of the model and the results in the general context of smart metering infrastructure, while in Section VIII we provide the final conclusions of the paper.

II. Advanced Metering Infrastructure & Data Tampering

A SG is a modern power grid enabling two-way power flow and bi-directional communication between power suppliers

The research was supported from ERDF/ESF ”CyberSecurity, CyberCrime and Critical Information Infrastructures Center of Excellence” (No. CZ.02.1.01/0.0/0.0/16_019/0000822).

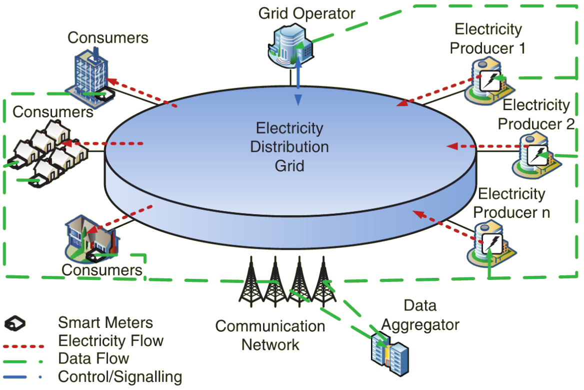


Fig. 1. Overview of the Advanced Metering Infrastructure (AMI) [5]

and consumers [2]. An efficient transmission of electricity, fast restoration in case of failures, and overall reduced costs for utilities are key aspects supported by the integration of cyber and physical systems [1]. The adoption of SGs leads to lower power costs for consumers, reduced peak demand, increased integration of large-scale renewable energy systems. Real-time monitoring and recovery of power generation and distribution is another key characteristic, as the actual state of the grid is monitored and reported to the network, adapting the power output to the real needs. SGs are also important to increase the usage of renewable sources (e.g., solar energy), as excess energy generated can be sold.

Decentralization of the SG led to the introduction of microgrids. A microgrid is an independent and small network of electricity users (consumers / prosumers) that can carry out operations independently from the centralized grid and even isolate itself from the rest of the power network in case of failure of the grid [6]. As it can be seen from Fig. 1, devices and sensors also play an important role in the context of SG, as they support smart energy scenarios, such as households using a solar-power system (with batteries and sensors) to decide about the best times to recharge Electric Vehicles (EV) [6].

Smart meters are a key element to allow bi-directional communication inside the AMI in SGs [7]–[10]. They constitute a cyber-physical device that can register power consumption and transmit back information to Distribution System Operators (DSOs). The smart meters allow one household to fully embrace the smart home concepts, by bringing several benefits to consumers / prosumers and DSOs: first of all, the availability of power consumption information allows consumers to make more reasoned choices about the best power consumption patterns allowing savings on energy costs. Furthermore, DSOs can remotely access smart meter readings, reducing the costs, and possibility of human mistakes. Additionally, wasting of energy can be reduced, by balancing the power needs where needed [8], [9]. The overall view of smart meters in the context of SGs can be seen in Fig. 2, where smart meters can be placed in the context of Home Area Networks (HAN) to integrate the different devices in a smart home. Furthermore, they can be part of Neighbourhood Area Networks (NAN) to integrate several households and make possible prosumer / consumer communication, and Wide Area Networks (WAN) to cover communication with data centers and DSOs. Data concentrators are an important component of the AMI that allows the connection between different smart meters and service providers [7], [9], [10]. All these connections and the way they are implemented, constitute the AMI [7], [9].

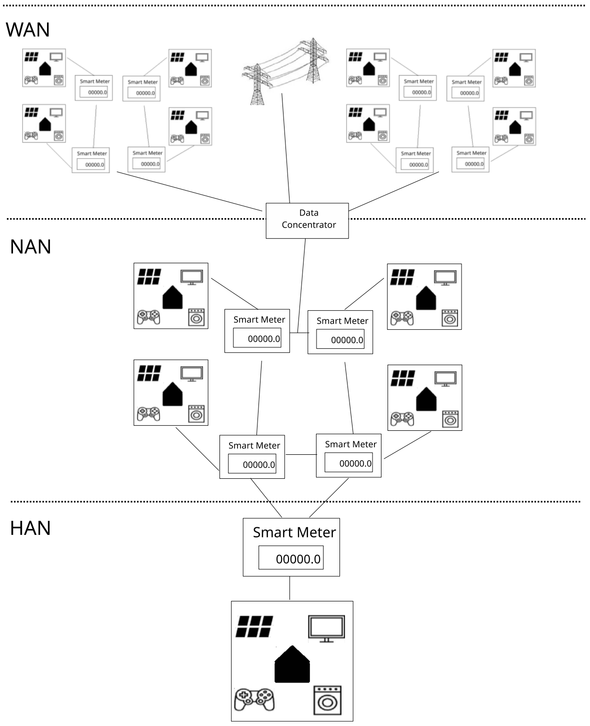


Fig. 2. Overview of Smart Meters and AMI infrastructure (adapted from [8])

Together with the benefits, there are also some potential drawbacks in the adoption of more advanced power metering devices and in general AMI. The large diffusion of smart meters and the enhanced functionalities offered increased the needs of a balance between securing the devices, keeping privacy concerns about data usage, and avoiding potential risks of energy theft.

Attacks to smart meters are often done with the aim of some data tampering activity: either on the physical device or on the data registered transmitted to achieve either some economical benefit or to harm the overall stability of the network. Data tampering activities are often referred to as false data injection attacks in the context of cyber-physical security of the SG. Attackers can change the smart meter measurements by either compromising the hardware devices locally, injecting false data packets sent to control centers or by changing data exchanged in other parts of the SGs infrastructure [3].

Data tampering activities targeted at AMI can be summarized in Table I, where we can see the cyber and physical attacks that can lead to some effects on power measurements reported by smart meters. Compromissions can be both derived from physical or cyber aspects connected to the AMI [11], [12]. In this paper, we are focused on the effects on power measurements, such as altering the reported power consumption to the energy provider.

III. Related Works

There are several research works that deal with data tampering in the AMI. Many of these papers focus on either energy theft detection, data tampering / data injection attacks, aggregation of data and / or frameworks for the detection / prevention of data attacks to smart meters in the AMI.

Li et al. [13] were proposing an approach to aggregate data from smart meters keeping privacy concerns in mind. A signature-based scheme, together with an incremental verification protocol is used to deal with potential dat tampering *in itinere* on data derived from the smart meters.

Hock et al. [14] were proposing an anomaly detection model using multiple sources to detect smart meters that were tampered with. The approach is based on the comparison of several time series, showing advantages over analyzing a single power consumption time series.

McLaughlin et al. [11], [12] proposed a framework for the detection of energy theft in the context of AMI, combining data from smart meters and sensors to increase detect capabilities. Combining power consumption data traces with data from logs and cyber events proved to increase the detection rate of malicious activities.

Liu et al. [15] use colored Petri net to model information flows between different components in the smart meters. Considering a threat model, authors propose a detection mechanism against false data injection attack that can be used to detect any data tampering activity.

The model proposed in the current paper can be seen as a temporal model that can be used on the inception of the data generated from smart meters. As such, it can be considered as an aggregation to the work done in Li et al. [13] rather than an alternative. Furthermore it can complement anomaly detection approaches (e.g., Hock et al. [14]) with some formal reasoning on which to base the algorithms for the detection of malicious activities for data tampering in the AMI.

IV. Model for smart meters data tampering detection

In this section, we will present the temporal logic model to be used as basis for algorithms for smart meters data tampering detection and processing of the generated data. In the upcoming sections we will discuss theoretical and practical proof of concepts of the model.

For any simple statements p, q, ..., any complex statements A, B, ..., the unary connectives $\neg$ (Negation), $\Box$ (Necessity), $\Diamond$ (Possibility), P (In the past), F (In the future), and the binary connectives $\wedge$ (Conjunction), $\vee$ (Disjunction), $\rightarrow$ (Entailment), the following recursive forming rules apply:

- (a) For any simple statement p, p is a well-formed statement. Furthermore, if $A = p$, then A is well-formed statement.
- (b) If A is a complex statement and $*$ is a unary connective, then $*A$ is a complex statement.
- (c) If A and B are complex statements and $*$ a binary connective, then $A * B$ is a complex statement.
- (d) There are no more statements than those defined by the clauses (a), (b) and (c).

By simple and complex statements we are referring to any kind of data that any device of the AMI might produce. In the current case we are focusing on the idea of implementing the model for data tampering on smart meters, but any reader would be able to identify statements that allow to reflect different aspects that give context to any action (e. g., weather data). Furthermore, while the model counts with a nice array of connectives, we have excluded any high order connectives (e. g., $\forall x$, for all x), as to keep the model to a minimum, therefore making its implementation easy as only simple operations would be required. Nevertheless, despite the simplicity of the model, it still allows for the processing of complex and interesting statements thanks to the expressiveness and variety of connectives. For example, for any reader that might be interested in using the model for patterns from a single smart meter could do so by using the recursive definition and add as many connectives to their statements as needed. Also, if a reader would be interested in aggregating multiple sources, the connective for Conjunction would allow for it. As an addition to the recursive definition of the connectives, we highlight that by $\top$ we mean constant true as customary.

A model M is the structure $M = \langle K, T, \models \rangle$, where K is the set of devices (smart meters) a, b, c, ...; i. e., $K = \{a, b, c, ..\}$; each element of K is a set in itself that includes a minimum and maximum power consumption, m and h respectively, among other characteristics i_1, i_2, i_3, ...; i. e., $a = \{m, h, i_1, i_2, i_3, ...\}$. T is a set of temporal points $t_1, t_2, t_3, ...$; i. e., $T = \{t_1, t_2, t_3, ...\}$. Finally, $\models$ is a relation from K to the set of statements such that the following clauses apply:

(1) $a \models A \wedge B$ if and only if (iff) $a \models A$ and $a \models B$
(2) $a \models A \vee B$ iff $a \models A$ or $a \models B$
(3) $a \models \neg A$ iff $a \not\models A$
(4) $a \models A \rightarrow B$ iff $a \models \neg A$ or $a \models B$
(5) $a \models \Box A$ iff $a \models m$
(6) $a \models \Diamond A$ iff $a \models h$
(7) $a, t \models PA$ iff $\exists s,\ s \in a$, with $s < t$, and $a, s \models A$, and $\forall u,\ u \in a$ if $s < u < t$, then $a, u \models A$
(8) $a, t \models FA$ iff $\exists s,\ s \in a$, with $t < s$, and $a, s \models A$, and $\forall u,\ u \in a$, if $t < u < s$, then $a, u \models A$

This model M is able to express multiple notions that are of use when considering data tampering in the SGs domain; specifically, it is based on the communication of power consumption values from the smart meters. In the first place, it is necessary to point out that the model is built under the idea that every smart meter can, and will, produce statements regarding their consumption. These statements are divided into two

TABLE I
TYPE OF DATA TAMPERING ATTACKS [11], [12]

Cyber	Physical	Effect on Power Measurements
Compromise meters through remote network exploit	Break into the meter	Stop reporting entire consumption
Modify the firmware/storage on meters	Reverse the meter	Remove large applicances from measurement
Steal credentials to login to meters	Disconnect the meter	Cut the report by a given percentage
Exhaust CPU/memory	Physically extract the password	Alter appliance load profile to hide large loads
Intercept/alter communications	Abuse optical port to gain access to meters	Report zero consumption
Flood the NAN bandwidth	Bypass meters to remove loads from measurement	Report negative consumption (act as a generator)

categories: simple statements, represented by lower-case letters p, q,..., and complex statements represented by upper-case letters A, B, C,...; simple statements are produced directly by the devices themselves while complex statements are to be obtained from the aggregation of simple statements. These statements are assigned either true or false according to the device where they are produced. $a \models p$ and $a \not\models p$ represent that the statement p is valid and not valid on the device a respectively. These statements are to be processed according to the classical propositional connectives of conjunction, disjunction, negation and entailment as customary. This is represented by clauses (1)-(4). (1), the clause for conjunction ($\wedge$, and), states that the conjunction of two statements is valid (in the device a) iff both statements are valid (in the device a). (2), the clause for disjunction ($\vee$, or), states that the disjunction of two statements is valid iff any of those two statements is valid. (3), the clause for negation ($\neg$, not), states that the negation of a statement is valid iff said statement is not valid. Finally, (4), the clause for entailment ($\rightarrow$, if...then...), states that a conditional statement is valid if any, the negation of the first statement or the second statement, are valid. Up to this point, the model is pretty straight forward and includes little to no novelty regarding customary processing of data.

The remaining clauses, (5)-(8), introduce the more interesting aspects. Clause (5), the clause for necessity, states that a necessary statement is true iff the argument of said statement is valid according to the minimum set by the device. That means that every device a would have a established minimum consumption m that would, in its turn, generate a statement A. This statement, therefore, is to be considered as necessarily valid, $\Box A$, iff it holds according to the minimum m. The same holds for clause (6), the clause for possibility, with the great difference that it is considered against the maximum set by the device, h.

Clause (7), the clause for "In the past", states that a statement is in the past iff there is a past temporary moment in which the statement was valid and for each temporary past moment between the first one and the present, the validity of the statement holds. This means that given a statement A is valid in a device a, in a temporary moment t, iff the statement is valid in a past temporary moment s and in the device a, and also for each temporary moment u, such that $s < u < t$, the validity of A holds in a. The same holds for clause (8), the clause for "In the future" with the great difference resides that the additional temporal moments s and u are set in the future and, therefore $t < u < s$.

All in all, this model allows us to establish a minimum consumption statement A that is to be necessary, $\Box A$, whose validity ensures that the data cannot be tampered giving back a value that is too low. Also, the model allows the establishing of a maximum that cannot be trespassed, $\neg\Diamond A$, that would be able to determine any tampering in the data consumption regarding the higher values. Both minimum and maximum are set outside of the boundaries of the formal model, as they are dictated by real-world actions, physical parts of the system (e. g., the maximum energy consumption established by contract). This further expands on the versatility of the model, as it can be set to almost anything that might be wanted, be it a simple numeric value, be it a range of values, with ease. Also, the model is not only able to consider and validate these examples, but also is able to track them along time, as it is able to determine not only if something is valid in the past or the future, PA and FA respectively, but also validate those statements according to very specific temporary points t, and therefore making the flagging of tampering much more precise. This is due to the facts that spikes over the maximum and under minimum are to be expected, but they cannot be validated for a long time.

To finalize this section there is a point that need to be addressed: the implicit comparison operator built in the validation of the statements. As m and h represent a numeric value and there are statements strictly linked to them, there has to be a comparison operator of sorts. Nevertheless, as it can be seen in the model above, the comparison operator does not exist. This is mostly due to the fact that the comparison can happen with disregard to this kind of operator: it happens but dealing on absolute values; i. e., instead of comparing two different values, it compares the validity of the statements with regard to a physically set boundary. This helps to keep the model as simple as possible, lowering its computational complexity and making it easier to implement.

A. On the relation of the proposed model with LTL

A really interesting point to be make is about the relationship of the model with LTL (Linear Temporal Logic). It could be argued that the presented model is, indeed, related to

LTL and that is, without a doubt, a correct interpretation, as both share the same kind of temporal dimension: a linear one. Nevertheless, the proposed model is more than just LTL. It should be regarded as a fragment of LTL plus an extension of said fragment; i. e., the fragment comprised of the connectives $\wedge$, $\vee$, $\rightarrow$, $\neg$, P and F, (excluding the connectives U and S) and extended with the modal connectives $\Box$ and $\Diamond$. Because this, the model is not introduced with relation to LTL, as that would be detrimental to its understanding. This reason is the same why LTL is presented as an individual model and not as just an expansion of classical propositional logic. The same could be said for any temporal or real-logics. Despite all this, any tools supporting specification and proving for existing temporal logics should be easily applicable to the common fragment of the temporal model that we have defined.

V. Proof of Concept

In this section, we will give two different proof of concepts. One based off a theoretical example, in which we will go over an ideal household a, and a practical example, for which we will use the data of UMass Smart* Dataset [4]. We begin with the theoretical one and will progress into the practical later.

To keep the proofs of concept as simple as possible we will provide simple examples with the breaching of a minimum/maximum for a long time by single datapoints. Nevertheless, the model is expressive enough to track patterns. For example, the reader might want to consider a case in which after a consumption over the maximum, the energy usage is back within the established limits and this happens up to three times with exactly the same energy consumption. This could be represented by the complex statement $(P\neg\Diamond r \rightarrow s)\wedge(P\neg\Diamond r \rightarrow s)\wedge(F\neg\Diamond r \rightarrow s))$ where $\neg\Diamond r$ is a consumption above the maximum and s a regular energy usage. As such, the model allows to design and apply customized patterns to fit the specificity of the AMI in which data tampering activities are to be detected.

A. Theoretical Proof of Concept

Let us consider a to be a small and regular household. As any household, this one has an upper limit on energy consumption at once established by the contract. This limit is established in h in the previous model (cf. clause (6) and its definition) and it is a simple statement p that equals to said upper limit; e. g., $p =$ ”the consumption is under 3kW”.

Similarly, a lower limit m (cf. clause (5) and its definition) is also established. This lower bound is not as easy to pinpoint as it is possible to not have clear data on it since its inception. Nevertheless, this problem might be solved with the advent of many smart devices, like fridges, washing machines and many other household items. These devices are expected to be able to convey their consumption in real time as statements. This would allow to calculate a minimum based on those devices that are to be running at all times; e. g., a fridge. All this items on their own should provide multiple statements regarding consumption that can be summed up in an aggregator before leaving the household; e. g., the fridge might produce $q =$ ”the consumption is 350W”, the electric heating might produce $r =$ ”the consumption is 1kW”; and therefore, the complex statement is to be $A = q\wedge r$. Obviously, this complex statement A is to be established in m as we pointed before.

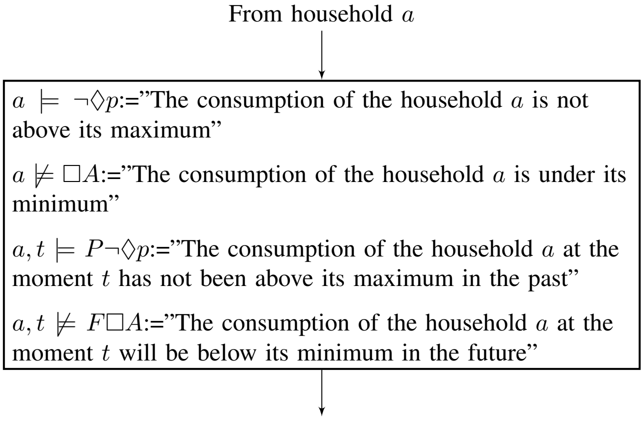
All this allows for monitoring peaks and lows in consumption. For that matter, clause (6) would allow to detect any peak higher that the maximum that we have established. When the statement $\neg\Diamond p$ is validated for the node a, $a \models \neg\Diamond p$, we know that the consumption data are being tampered, as it is impossible for the consumption to be as high: the complex statement $\neg\Diamond p$ indicates that is impossible for that to happen. In the same vein, any time that the statement $\Box A$ is not validated, $a \not\models \Box A$, it indicates that the consumption has gone under a minimum that is not expected, as there is minimal consumption that should happen constantly. With these in mind, we could track the peaks and valleys of the ideal household a that we have defined above, being able to set off an alarm when the consumption goes into abnormal territory.

It is obvious that peaks and valleys are bound to happen from time to time and not all of them should be due to data tampering. For that matter, the model introduces clauses (7) and (8) that allows to track the abnormal consumption as time goes by. The previous statements could be modified so they are read as $P\neg\Diamond p$ and $F\Box A$. This means that the abnormally high consumption from before has been going on for some time. Even more, the validation of this statement in the household that we have set should be reading as $a, t \models P\neg\Diamond p$, indicating that since the time point t the consumption has been too high. This would be able to tell us that the data of household has been tampered if t is far away enough in time. In the same sense, the not validation of $F\Box A$ would be the anticipation of some tampering in the long run: $a, t \not\models F\Box A$ means that the abnormally low consumption $\Box A$ is being constantly validated. This, in particular, rather than establishing the revision of something that has been happening for some time to set off an alarm, would be useful to indicate in which time point the alarm should be set off.

All the statements that we have used for this theoretical example can be seen summarized in Figure 3. Also, as this figure points out, the statements that are outputted by the household are to be processed in some way or another, be it a semantic web reasoner as mentioned above, be it a manual processing. One thing might draw the attention of the reader from the diagram is the explanation of $a, t \not\models F\Box A$. This is due to the fact that we are predicting what will happen in the future. Nevertheless, what this mean is that said statement is able to track the minimum consumption into the future; i. e., the statement allows for flagging a too low consumption somewhere in the future.

B. Practical Proof of Concept

Let us consider the case of HomeA from the UMass Smart* Dataset (2017 release) [4]. As we have stated before, these data are extracted directly from real smart meters and real households, so this further validation reinforces the usefulness and of the model and its real-world application. For this proof



From household a

$a \models \neg\Diamond p$:="The consumption of the household a is not above its maximum"

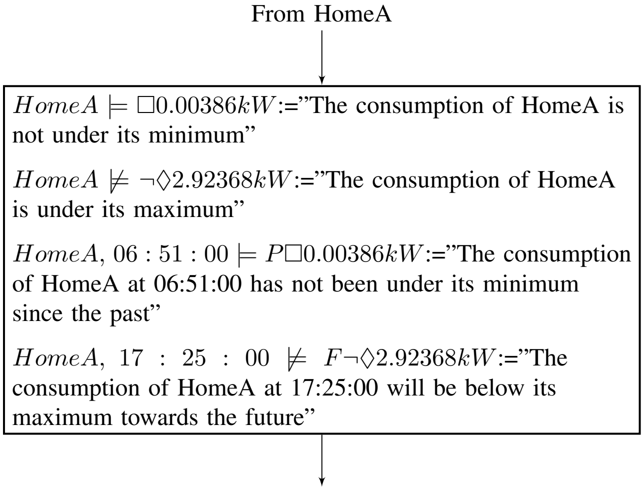
$a \not\models \Box A$:="The consumption of the household a is under its minimum"

$a, t \models P\neg\Diamond p$:="The consumption of the household a at the moment t has not been above its maximum in the past"

$a, t \not\models F\Box A$:="The consumption of the household a at the moment t will be below its minimum in the future"

To the processing of data

Fig. 3. Household a Data Diagram



From HomeA

$HomeA \models \Box 0.00386kW$:="The consumption of HomeA is not under its minimum"

$HomeA \not\models \neg\Diamond 2.92368kW$:="The consumption of HomeA is under its maximum"

$HomeA$, $06:51:00 \models P\Box 0.00386kW$:="The consumption of HomeA at 06:51:00 has not been under its minimum since the past"

$HomeA$, $17:25:00 \not\models F\neg\Diamond 2.92368kW$:="The consumption of HomeA at 17:25:00 will be below its maximum towards the future"

To the processing of data

Fig. 4. HomeA Preexisting Data Diagram

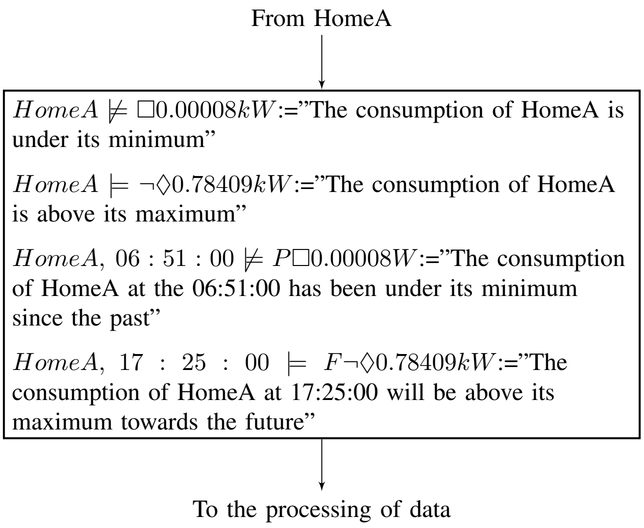
of concept, we will divide it into two different parts. The first one would be focused on showing how the model works with the preexisting data, while the second will focus on a hypothetical data injection attack.

1) Preexisting Data: For this example, we focus on the data stored in the file HomeA-meter3_2016, the one with the meter reading occurring every minute. Despite being real-world data, we are still missing some important values, like the minimum and maximum, m and h in the model respectively. Therefore, we will extrapolate this from the data we already have. As we have pointed above, we are considering single datapoints for the minimum and maximum but, nevertheless, it could be adapted to be a value equal to a standard deviation over the mean as we will show later. Since we are dealing with a regular household we will assume that there is no data tampering in the dataset and, from the file, we can assume that $m = "0.00010kW"$ and $h = "3.50000kW"$. These values are obtained from the data: there is no value under 0.00010kW as the lowest consumption can be found at Date: 2016-08-02, Time: 15:09:00 with the consumption equal to 0.00013kW; also, there is no value above 3.50000kW as the highest consumption can be found at Date: 2016-05-09, Time: 17:38:00 with the consumption equal to 3.14308kW.

Once we have set the upper and lower bound we introduce the operators of the model that allow for the description of minimum and maximum. For the case of the minimum, for the consumption p, where p is the consumption of the data from Date: 2016-10-07, Time: 06:51:00 and is $p = "0.00386"$. Then we have that $HomeA \models \Box p$, as p remains over the minimum we have set. For the case of the maximum, for the consumption q, where q is the consumption of the data from Date: 2016-05-13, Time: 17:25:00 and is $q = "2.92368"$. Then we have that $HomeA \not\models \neg\Diamond q$, as q remains below the maximum set in h. For the case of the time-sensitive connectives is easy to see how they are implemented from the time points that we have selected. Assuming that we want to check out the validity of the minimum consumption, $\Box p$, in the past, $P\Box p$, we would set the time point t as $t = "06:51:00"$ and go as back we might be interested, in this case we set $s = "04:51:00"$ for a time span of two hours. Since for every time point u between t and s the consumption does not go under the minimum, as it can be seen in the dataset, we can affirm that $HomeA, t \models P\Box p$; i. e., the consumption of HomeA has not gone under the minimum in the past at the time point t (since a time point s). Additionally, for the case of checking the maximum in regards to a future time point, we firstly set the time point t' as $t' = "17:25:00"$ and the future time point s' as $s' = "21:30:00"$ giving a time span of 4 hours and 5 minutes. Since for every time point u' between t' and s' the consumption is under the maximum we know that $HomeA, t' \not\models F\neg\Diamond q$; i. e., the consumption of HomeA has not gone over the maximum at time point t' (towards a time point s').

All that has been described for this practical proof of concept based on the preexisting data can be seen in Figure 4.

2) Simulated Data Injection Attack: Now, we will show how the model works in the case that a data injection attack might happen at HomeA. For this case, we will consider two different data injection attacks and, to showcase the flexibility of the model, we will set up the maximum to be equal to the mean plus three times the standard deviation. In this case the mean is 0.02166kW while the standard deviation is 0.21316kW. The minimum would be set up to be a really low value, as the mean minus three times the standard deviation would give back a negative value, something that is impossible in this case. Obviously, this statistic approach will not work appropriately given the fact that the dataset that we are using is not normally distributed, but will suffice to show how the model works. With this in mind the minimum and maximum are as follows: $m' = "0.00009kW"$ and $h' = "0.66114kW"$. The first one would consider that the energy consumption p' at Date: 2016-07-10, Time: 04:51:00 until Time: 06:51:00



$HomeA \not\models \Box 0.00008kW$:="The consumption of HomeA is under its minimum"

$HomeA \models \neg\Diamond 0.78409kW$:="The consumption of HomeA is above its maximum"

$HomeA$, $06:51:00 \not\models P\Box 0.00008W$:="The consumption of HomeA at the 06:51:00 has been under its minimum since the past"

$HomeA$, $17:25:00 \models F\neg\Diamond 0.78409kW$:="The consumption of HomeA at 17:25:00 will be above its maximum towards the future"

Fig. 5. HomeA Simulated Data Injection Attacks Diagram

has been tampered so it reads $p' = "0.00008kW"$. Since we know that the minimum $m' = "0.00009kW"$, it automatically follows $HomeA \not\models \Box p'$, thus flagging the consumption at any time point given of that time span as abnormal. Furthermore, since the consumption have been modified for time span, from $s = "04:51:00"$ to $t = "06:51:00"$, it allows the model flag it as abnormal in the current time point t with regards to a time point in the past s; i. e., $HomeA, t \not\models P\Box p'$. All in all, the model would detect that, at the time point t, the energy consumption has been abnormally low since the time point s.

The second simulated data injection attack would take place at Date: 2016-05-13, Time: 17:25:00 until Time: 21:30:00, tampering the energy consumption q' to $q' = "0.78409kW"$. Since the maximum has been set as $h' = "0.66114kW"$ this would allow for the model to give back the flagging of the consumption as abnormally high at any time point of the previous time span; i. e., $HomeA \models \Diamond\neg q'$. Furthermore, if we consider the time point $t' = "17:25:00"$ as the initial time point, the model would be able to track this towards a future time point $s' = "21:30:00"$ and, thus it would mark the consumption as abnormally high for the time span since t' until s'; i. e., $HomeA, t' \models \Diamond\neg q'$.

These simulated data injection attacks are available in the shape of a diagram to the reader in Figure 5. Additionally, it is important to mention that these attacks come to show how the model would work with a high tolerance, as it would detect really small differences, like in the case of the minimum, while also detecting some not so small such as the attack on the maximum.

VI. Formal Results

The fragment M_0 of the model M that includes the connectives $\wedge$, $\vee$, $\neg$, $\rightarrow$, P, F is sound, complete, decidable and satisfiable. This is due to the first four connectives being the connectives of classical propositional logic. For the last two, the temporal connectives this results also apply. This is due to the fact that the temporal dimension has been added following the work done in [16], which makes them a conservative extension of the previous model. With this said, they necessarily preserve any properties that the base model might have. Therefore, the whole fragment is, as pointed above, sound, complete, decidable, and satisfiable. This formal results guarantee that the model will not get stuck in an infinite loop, that it would be able to process any valid statement no matter what, and also the fact that well-formed statements can be validated by the model.

Also, it should be recalled that the temporal dimension added following [16] is a Linear Temporal Dimension. This implies that there is just one flow of time, not multiple as in the case of a Branching Temporal Dimension. This further expands on the simplicity of the model as, while the branching time option could be really interesting, also requires more computational power, as it creates a different flow of time for each event that we might want to track thanks to the model.

With regards to the missing fragment, the one of the connectives $\Box$, $\Diamond$, the same result should follow, but for that matter the model should be strengthened with a relational operator R. Since this model aims at being a simple model of low computational complexity, this relation should be avoided. Nevertheless, since the main validation terms of the connectives, m and h, are expected to be based on real-life events instead of theoretical ones, the same results should follow, but their proofs exceed the capabilities of a formal system.

VII. Discussion

The proposed model has been developed with the aim to deal with smart meters data tampering potentially being the basis for algorithms for anomaly detection and a semantic web reasoner. However, it is not only ideal to prevent the submission of false data, it can also constitute a validator to process regular generated data from the AMI. The model would allow for service providers to keep the whole network under surveillance to further support additional data monitoring processes. Furthermore, the definition of the model is as minimal as possible so its real-world implementation is not huge tax on any preexisting running process. Also, the fact that the model has been endowed with a temporal dimension helps when dealing with questions that might fall outside the scope of other processing tools. This is even more evident when compared with the data that is already available, like the one of UMass Smart* Dataset [4] that we used in the practical proof of concept.

A point that needs to be addressed is the integration of the model with the semantic web and therefore, with the semantic web reasoners. The statements that are part of the model are considered to be as statements from IoT devices. Generally speaking, this means that these statements are easily converted into semantic web statements, making them processable by any kind of semantic reasoner. This conversion from IoT into the semantic web is due to an Internationalized Resource

Identifier (IRI) that gives uniquely identifiable names to the thing and also specifies the location of the resource, based on the Web Ontoly Language (OWL). All in all, the model is not introduced only with the idea of detecting data tampering, but also to implement an automation of so and the AMI in general. This can be more obvious if one takes a look at all the clauses of the model, as it might be possible to have a model with only (3) and (5)-(8) to process the data tampering, but in this case, the model is extended so a semantic web reasoner has the possibility of going beyond just data tampering. It is interesting to mention that there are semantic web reasoners that have been already developed with the idea of working the energy consumption data that is the base of SGs. A good example of what these can reasoners do is [17], where the authors introduce the OEMA ontology for unifying the energy domain, which leads to the automation of the energy performance and contextual data processing. Let us add that the proposed model is independent of any ontology, it can be used to work with many different ones, as it provides the framework for the reasoning, rather than the statements that can be processed and, therefore, adding to its flexibility.

It is also worth mentioning how the model flexibility is one of its main advantages. As the model is quasi-formal, meaning that there is a part based on its interaction with the physical world, the constraints established can be bent in whatever way a service provider of a SG might need. For example, when establishing the maximum and minimum consumption for any prosumer, the aforementioned methods might not work. This is due to the fact that the minimum consumption of a prosumer might, and is expected to be, negative. Nevertheless, this can be extrapolated from the data of the energy production of the solar panels. Thus, modifying the validity of m and h, the formal notions of the model that represent the minimum and maximum consumption of the element a of the network.

Going further into the flexibility of the model, for two different households a and b, if those two households have the same minimum and maximum, m and h, then the model would be able to assign the corresponding statement to each household. Furthermore, in the case that the minimum and maximum are different, the model is not just THE model, but rather A model, meaning that there could be multiple iterations of the same model for different entities, but with the same structure. Additionally, given that both m and h are determined by the iteration of the model, the same semantic web reasoner could be used to process the data of multiple models at the same time.

To conclude this section, it is important to note how the time-sensitive aspect of the model, does not relay on accessing time points that have not already happened; i. e., time points in the future. Rather than that, what the model offers is the option of tracking changes with the passing of time, i. e., the F connective, or checking with past time points, i. e., the P connective, to ensure the validity of the statements that are happening in the current time point.

VIII. Conclusion

This paper has introduced a time-sensitive model that allows for the detection of anomalies in energy consumption from smart meters in the context of data tampering activities. The model not only offers the detection of said anomalies, but also their tracking along the time dimension, allowing for the flagging of irregularities that are sustained in time. This model has been shown to be able to detect any case of data tampering in smart meters, as it would not automatically target any peak or valley in the consumption, but rather those that prolong their existence over time. The effectiveness of this very model has been shown through a proof of concept, at first theoretically and based on a real dataset afterwards. Furthermore, the model can be taken as the basis for the implementation of a semantic web reasoner that is not just focused on data tampering, but also allows for processing any other information produced by the smart meters that might be part of the whole advanced metering infrastructure.

There are multiple lines of investigation that can be followed from here; the main ones to be explored include the implementation of an ontology and a semantic web reasoner based upon the model described. Together with this support, the data-tampering detection model described would be tested within an anomaly detection framework, thus allowing more data to be obtained for further validation. The model also could be implemented in different domains like the communication solution that appeared in [18]. It is also of interest to modify the model so the temporal dimension may be changed from a linear one to a branching one, thus allowing for the tracking of multiple time spans of different households at the same time.

References

[1] S. Goel, Y. Hong, V. Papakonstantinou, and D. Kloza, *Smart grid security*. Springer, 2015.

[2] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart grid—the new and improved power grid: A survey," *IEEE communications surveys & tutorials*, vol. 14, no. 4, pp. 944–980, 2011.

[3] S. Aoufi, A. Derhab, and M. Guerroumi, "Survey of false data injection in smart power grid: attacks, countermeasures and challenges," *Journal of Information Security and Applications*, vol. 54, p. 102518, 2020.

[4] "Smart - UMass Trace Repository." [Online]. Available: http://traces.cs.umass.edu/index.php/Smart/Smart

[5] Z. Erkin, J. R. Troncoso-Pastoriza, R. L. Lagendijk, and F. Pérez-González, "Privacy-preserving data aggregation in smart metering systems: An overview," *IEEE Signal Processing Magazine*, vol. 30, no. 2, pp. 75–86, 2013.

[6] R. Hebner, "Nanogrids, microgrids, and big data: The future of the power grid," *IEEE Spectrum Magazine*, p. 23, 2017.

[7] S. Chren, B. Rossi, and T. Pitner, "Smart grids deployments within eu projects: The role of smart meters," in *2016 Smart Cities Symposium Prague (SCSP)*, May 2016. doi: 10.1109/SCSP.2016.7501033. ISSN null pp. 1–5.

[8] D. B. Avancini, J. J. Rodrigues, S. G. Martins, R. A. Rabêlo, J. Al-Muhtadi, and P. Solic, "Energy meters evolution in smart grids: A review," *Journal of cleaner production*, vol. 217, pp. 702–715, 2019.

[9] S. S. S. R. Depuru, L. Wang, V. Devabhaktuni, and N. Gudi, "Smart meters for power grid—challenges, issues, advantages and status," in *2011 IEEE/PES Power Systems Conference and Exposition*. IEEE, 2011, pp. 1–7.

[10] G. R. Barai, S. Krishnan, and B. Venkatesh, "Smart metering and functionalities of smart meters in smart grid-a review," in *2015 IEEE Electrical Power and Energy Conference (EPEC)*. IEEE, 2015, pp. 138–145.

[11] S. McLaughlin, B. Holbert, S. Zonouz, and R. Berthier, "Amids: A multi-sensor energy theft detection framework for advanced metering infrastructures," in *2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm)*. IEEE, 2012, pp. 354–359.

[12] S. McLaughlin, B. Holbert, A. Fawaz, R. Berthier, and S. Zonouz, "A multi-sensor energy theft detection framework for advanced metering infrastructures," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 7, pp. 1319–1330, 2013.

[13] F. Li and B. Luo, "Preserving data integrity for smart grid data aggregation," in *2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm)*, 2012. doi: 10.1109/SmartGridComm.2012.6486011 pp. 366–371.

[14] D. Hock, M. Kappes, and B. Ghita, "Using multiple data sources to detect manipulated electricity meter by an entropy-inspired metric," *Sustainable Energy, Grids and Networks*, vol. 21, p. 100290, 2020.

[15] X. Liu, P. Zhu, Y. Zhang, and K. Chen, "A collaborative intrusion detection mechanism against false data injection attack in advanced metering infrastructure," *IEEE Transactions on Smart Grid*, vol. 6, no. 5, pp. 2435–2443, 2015.

[16] M. Finger and D. M. Gabbay, "Adding a temporal dimension to a logic system," *Journal of Logic, Language and Information*, vol. 1, no. 3, pp. 203–233, Sep. 1992. doi: 10.1007/BF00156915. [Online]. Available: https://doi.org/10.1007/BF00156915

[17] J. Cuenca, F. Larrinaga, and E. Curry, "A Unified Semantic Ontology for Energy Management Applications," in *WSP/WOMoCoE@ISWC*, 2017.

[18] P. Hajder, M. Hajder, M. Liput, and M. Nycz, "Direct communication of edge elements in the industrial internet of things," in *Communication Papers of the 2020 Federated Conference on Computer Science and Information Systems*, ser. Annals of Computer Science and Information Systems, S. Agarwal, D. N. Barrell, and V. K. Solanki, Eds., vol. 23. PTI, 2020. doi: 10.15439/2020KM194 pp. 35–42. [Online]. Available: http://dx.doi.org/10.15439/2020KM194

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 521–531

DOI: 10.15439/2021F81
ISSN 2300-5963 ACSIS, Vol. 25

An Agent-based Cyber-Physical Production System using Lego Technology

Metehan Mustafa Yalcin*, Burak Karaduman†, Geylani Kardas‡ and Moharram Challenger§
*Department of Electric and Electronics Engineering, Ege University, Izmir, Turkey
metehanmustafayalcin@gmail.com
†Department of Computer Science, University of Antwerp and Flanders Make, Belgium
burak.karaduman@uantwerpen.be
‡International Computer Institute, Ege University, Izmir, Turkey
geylani.kardas@ege.edu.tr
§Department of Computer Science, University of Antwerp and Flanders Make, Belgium
moharram.challenger@uantwerpen.be

***Abstract*—To cope with the challenges of constructing Cyber-physical Production Systems (CPPS), many studies propose benefiting from agent systems. However, industrial processes should be mostly emulated while agent-based solutions are integrating with CPPS since it is not always possible to apply cyber-based solutions to these systems directly. The target system can be miniaturised while sustaining its functionality. Hence, in this paper, we introduce an agent-based industrial production line and discuss the system development using Lego technology while providing integration of software agents as well as focusing on low-level requirements. In this way, a CPPS is emulated while agents control the system.**

***Index Terms*—Software Agent, Multi-agent System, SPADE Agent Programming, Cyber-Physical Production System, SysML**

I. Introduction

ADVANCES in networked systems produce new paradigms and new design challenges in the embedded systems. The information processing and computation are merged with communication and control that creates Cyber-Physical Systems (CPS) [1]. This evolution expands the capabilities of embedded technology interacting with the physical world through computation, control and networked communication. In this way, medical devices, transportation vehicles, intelligent highways, robotic systems and factory automation can be instrumented and implemented considering new capabilities that are achieved by CPS. One of the specialized fields of CPS is the Cyber-physical Production Systems (CPPS) which is related to the autonomous and cooperative elements and subsystems that are connected based on the context within and across all levels of production, from processes through machines up to the production and logistics networks [2].

Smart manufacturing considers adapting the embedding software and hardware technologies to the CPS, including intelligent methodologies. It aims at increasing the efficiency in the production as well as improving the conditions in the delivery process. Moreover, it is one of the leading application domains since it can have large scale production in domestic and international marketing that can impact highly economic growth. Industry 4.0 takes a pioneering role to determine manufacturing standards of the future [3]. A highly challenge in manufacturing came forward is flexibility since there are high demands for products. It is very problematic to meet those demands because of safety and complexity that arise from frequent interactions and co-operative requirements between machines, lack of human experts, and absence of an intelligent mechanism that can reason unpredictable behaviours of the system [4]. However, the requirement for intelligence to achieve smart CPS has emerged due to the complexity of these systems and physical unpredictability.

To cope with the challenges of CPS, many studies propose benefiting from the features of multi-agent systems (MAS) (e.g. [5]–[7]). MAS are widely preferred for providing support for smartness, decentralization, autonomy, and socialization of CPS. They increase the effectiveness of CPS providing enhanced functionalities for production and automation. The software agents can decide reconfiguration of the control functions/parameters, monitor transition between processes, and observe the human errors while increasing the system/human safety. Moreover, they can detect module breakdowns, structural changes, and contradictory inputs and materials, then they plan and decide on a suitable solution. In this way, they can enhance product quality and prevent damages during critical processes.

An integration of MAS and CPS may facilitate the use of intelligent agents in various industrial applications [8]. Once agents can control the components of the CPS, the developer can focus on higher-level solutions such as implementing intelligence mechanisms [9], aggregating Big Data and creating Digital Twins [10]. However, industrial processes should be emulated while agent-based solutions are integrating with CPS to address its challenges. Because it is not always possible to apply cyber-based solutions to the operational systems and dangerous environment of the industry directly when requested. Moreover, it is a burden to prototype an actual industrial production system for development purposes. Therefore, the target system can be miniaturised while sustaining its functionality, accuracy and goal-orientedness.

Firstly, a composable and concrete technology to mimic the industrial systems where CPSs are intensively operational is required. One of the technologies commonly used for imitating such systems is Lego (e.g. [11], [12]). Although Lego technology can be supported with tools or languages such as *Scratch* [13] for programming its hardware devices to control motors and collect data from sensors, it is not possible to integrate software agents and any intelligence mechanism easily. Secondly, a common development environment and language is required to merge Lego technology and agent software. Lastly, the integration should be seamless and built from scratch, and the system should behave as it is developed by the *Scratch* graphical programming language. Hence, in this study, we introduce an agent-based industrial production line and discuss the design and implementation of this system using Lego technology while providing the integration of the software agents both to address the abovementioned CPS problems and to focus on low-level requirements.

Since CPPS use different controller/computation parts, their relationship should be modelled to reduce their development and design complexity [14]. For this purpose, the analysis and design of both the software and system parts are realized using SysML [15] in our study. Physical implementation is done using Lego technology and Raspberry Pi (with PiStorms hat) while embedded and agent software is coded using Python and Smart Python Agent Development Environment (SPADE) [16], respectively, including RasberryPI-Lego library [17]. This paper discusses all these parts of the system development. In addition, the challenges during the integration of Lego technology and software agents to create a CPPS are discussed, and lessons learned are also given in the paper.

The rest of this paper is organized as follows: Section 2 briefly discusses the related work. Section 3 gives the analysis and design of the smart production line system. The implementation of the software components and the setup of the hardware are all discussed in Section 4. The challenges we faced and lessons learned are reported in Section 5. Finally, the paper is concluded and the future plan is described in Section 6.

II. Related Work

Multi-agent systems are broadly researched and developed for providing modularization of the dynamic systems, decentralization for distributed systems [18], autonomy for production, and re-usability for further development of physical systems [19]. However, before realising such complex operations, agent integration has to be provided [20]. Once agents are implemented into CPS, their control over embedded functions should also be ensured.

In [7], capabilities of agents and CPS challenges are matched while underlining the software agents are generally a good fit for the requirements of the next generation CPS. Therefore, agents can show paramount effects for creating collaboration and integrity when they are distributed, providing smart decisions when physical unpredictability exists during the operation of CPPS. Leitao et al. [7] also emphasize that agents are good at reasoning e.g. using machine learning techniques, providing sustainability and managing human interaction in CPS.

The study in [21] addresses joint characteristics of Industrial Internet-of-Things (IIoT) and CPS while it also provides methodologies about the applicability of IoT-enabled solutions to CPPS considering interoperability principles. Additionally, they also present modelling approaches for IIoT systems.

In [22], the association between CPS and Embedded systems is considered. It is suggested to use a micro-controller board with various communication interfaces such as CAN, UART, WLAN, Ethernet, and BLE. In this way, this micro-controller can provide system-level compatibility with various boards and technological diversity to extend the design space.

Lee [23] discusses the design challenges of CPS in general from various perspectives and proposes a model-based design as a complementary approach. Hence, the process of rewriting the CPS software every time for each system can be shortened or even eliminated.

Similarly, the application of zero defect manufacturing using software agents is studied in [24] to cope with the challenges of CPS in the smart manufacturing domain. The researchers create a four-layer architecture and benefit from IoT solutions to inter-operate it with CPS using an edge-fog-cloud methodology. They highly consider earlier detection of anomalies, product quality and data correlation to find the optimal solution without interfering with any control functions.

In [25], an agent-oriented system is proposed for an Automated Guided Vehicle (AGV) with the on-board camera. Xing et al. [25] benefit from the MAS paradigm to provide an effective organisation and communication between system components. They indicate that the MAS paradigm improves the intelligence of the systems by providing an onboard solution while achieving context-awareness for an autonomous AGV.

Queiroz et al. [26] discuss the cognitive requirement of CPS, exhibit the necessity of the distributed intelligence, and envision the usefulness of MAS as they fit the CPS. They indicate that autonomous decisions in a decentralised way can address some of the CPS challenges.

In [27], an ontological classification of CPS is made considering past, present, and future CPS technologies emphasising the requirement of intelligence. Moreover, intelligence level and self-* features of CPS are matched considering both the previous achievements and future projections. The study also focuses on the current research gaps in this domain.

In [6], the authors suggest using an agent development platform, called Tartarus, to implement both cyber-physical and IoT systems. They use a solution to run the software agent on the Intel Galileo and RaspberryPI boards using the Tartarus-Lego Mindstorms NXT robots programming interface. Although the current study also supports our vision to achieve agent-CPS integration using Lego development components, our solution differentiates in the sense that we focus more on integrating agent behaviours with the low-level of embedded control of the system components. This refers to low-level problems of agent-CPS integration from the bare-metal embedded libraries to binding them with agent-based programming.

Petrovska et al. [28] propose a domain-independent approach for knowledge aggregation and reasoning of decentralized monitoring in multi-agent smart CPS. According to their logic algorithm, they tackle the uncertainty of partial, faulty and potentially conflicting context observations. Their approach allows capturing uncertainty at run-time on a local level while providing a global decision-making mechanism. They evaluate their approach using multiple rooms cleaning robots implementing MAPE-K feedback loop to their multi-robot system.

The study in [5] discusses how a domain-specific modeling language, called SEA_ML++ and its tool [29], [30] are used for the design and implementation of a cyber-physical garbage collection system. The system is first modelled according to SEA_ML++'s graphical concrete syntax. Then a significant portion of the agent-based implementation of the system is automatically generated from these models via a series of model-to-code transformations.

In [31], the use of agents on Raspberry Pi is introduced. The study mostly focuses on the networking of agents and the cyber part of their location-aware and tracking services to establish an indoor person tracking system.

The survey in [32] considers the state of the art of applying agent technologies into the industry. The authors indicate that the industrial systems should be coupled with software logic and software agents to design CPS. They also underline the integration of software agents with physical hardware is both a difficult and a long-term process, and hence the common software patterns and paradigms can be applied to construct industrial agents which control the industrial machines and devices. However, according to their results, there is no uniform way to integrate the software agents to the low-level automation functions to create the industrial agents. Our methodology, which will be discussed in the following sections of this paper, may provide a strong alternative on facilitating the related integration within this perspective, specifically by emulating the industrial system before the real implementation and benefiting from both the agents and the embedded software and hardware.

Karnouskos et al. [33] classify the industrial agents according to ISO/IEC SQuaRE standards [34] under 8 categories, namely *Usability*, *Compatibility*, *Performance Efficiency*, *Functional Suitability*, *Portability*, *Maintainability*, *Reliability* and *Security*. Considering these 8 categories, an industrial system can be mimicked, and these standards can be applied to test the validity of them before the developed methodologies are adapted to the actual system.

As can be seen, while most research in the literature focuses on providing intelligence, adaptiveness and awareness mechanisms for CPS using agent technologies from a higher level of view, our study contributes to these efforts by providing an underlying infrastructure to merge embedded software with agent programming as well as mimicking the system operations over Lego technology to achieve the physical emulation of the industrial-like systems before their construction. Thus, we believe that once such an infrastructure is provided, then applying high-level solutions via decision making, knowledge extraction or pattern matching as mainly considered in the current studies can become more feasible.

III. System Software Analysis & Design

In this section, we discuss the analysis and design of our smart manufacturing system using SysML. We provide a multi-agent, multi-layered, multi-process study for such manufacturing systems. At the cyber side, the scalability, reactivity, and communication are merged with the embedded software in order to control a composable, extensible and modular Lego-based physical system.

A. System Overview

During the analysis and design, an efficient, autonomous, and smart manufacturing system is aimed to emulate the industrial requirements and tasks. The different types of input products are sorted in this system and they are processed autonomously according to their features which are similar to the common functionalities in an industrial factory.

The operation of the production line starts from inputting Lego bricks into the system. Then, the system starts to deliver these bricks using conveyor belts and in the next phases, the system decides either to sort or to combine these Lego bricks according to their colours.

The system is represented by a block diagram, which is illustrated in Figure 1, to provide an overview of the design. Considering the achievement of an autonomous and a modular system, the system is designed to be working on two embedded devices which are represented as *layer 1* and *layer 2*. The essential requirement to run the whole system is the agent communication which is established between these two layers using XMPP protocol [35]. Two layers controlled with two *PiStorms* extension boards and two RaspberryPI3. The first layer controls 4 motors, 1 button, 1 ultrasonic sensor and 2 colour sensors while the second layer controls 3 motors and a limit switch.

Each software agent (shown in the photograph of the created system in Figure 2) has its own tasks and roles inside the subsystems of the production line. In the following subsections, they are discussed in detail. First of all, each agent has specific behaviours and actions to control hardware elements. These actions provide the sustainability to make the system complete its processes successfully. While four of seven agents work with cyclic behaviour, two agents have one-shot behaviour and an agent works based on a finite state machine (FSM) behaviour. To get the system and the agents ready, "Initialize" methods of all agents are triggered at first. Agents act based on their roles. The roles of the system agents are as follows: Drop agent is responsible for delivering products from system input to the Shredder agent. Shredder agent is responsible for shredding products and delivering them to Sort agent. Sort agent should decide about the product and move it to a related process. Push agent removes the brick from the conveyor belt. Lastly, Build agent builds required products according to the current state of its FSM behaviour model. Collaboratively, all agents run and control the whole production process.

In this regard, agents execute their programmed behaviours to achieve their goals. Before they start executing their tasks, each agent awaits a message from the preceding agent. This communication system provides a proper sequence for agent executions in the system.

B. Architectural Design

We designed the system architecture using block definition diagrams. For instance, in Figure 1, hardware layers are represented with root classes, named *dev1* and *dev2*. These classes are specialized to assign specific functions for the goals of agents. These classes are created using *Singleton Pattern* to constraint the instance creation as only one instance per *PiStorms* device. We benefit from *PiStorms* library to program the device-specific features and the functions which are used by the *dev1* and *dev2* classes. Software agents control the hardware I/O ports via these singleton classes. These classes constraint the cardinality of object creation to one for each hardware element and these device objects are accessed by software agents to use device functions for I/O operations. In this way, agents control the device I/O to achieve their goals and sustain the operation of the production line.

C. Agent Communications

In a MAS, messaging is important for agents to complete tasks collaboratively. In SPADE, Agent Communication Language (ACL) messages have various parameters and commonly used ones are *type*, *receiver*, *sender*, and *content*. In our system, we use informative messages to establish organization between agents. When certain events occur in the system, agents send messages which include keywords (performatives) and lead triggering an action inside the agent receiving that message. SPADE uses the XMPP protocol to deliver messages and to ease connection creation. The sequence diagram given in Figure 3 represents the messaging between the system agents.

D. Behavioural Design

In this section, behavioural activities of the agents (emulating the product line robots) are discussed. As illustrated in Figure 4, each agent has specific behaviours and actions to control hardware elements.

Overall, while four of seven agents work with cyclic behaviour, two agents have one-shot behaviour and an agent work based on finite state machine (FSM) behaviour. These agents provide actions for the sustainability of the system. Moreover, the process transitions, controlled by the software agents of the system can be visualized as given in Figure 5.

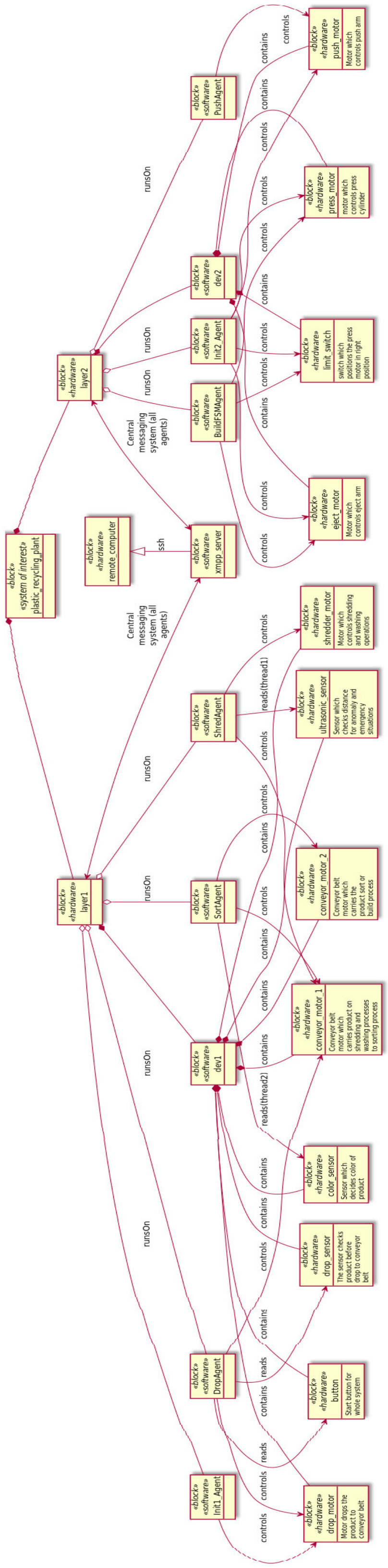

«block» «system of interest» plastic_recycling_plant
«block» «hardware» layer2
«block» «hardware» layer1
«block» «hardware» remote_computer
«block» «software» xmpp_server
ssh
Central messaging system (all agents)
Central messaging system (all agents)
runsOn
runsOn
runsOn
runsOn
runsOn
runsOn
«block» «software» PushAgent
«block» «software» dev2
«block» «software» Init2_Agent
«block» «software» BuildFSMAgent
«block» «software» ShredAgent
«block» «software» SortAgent
«block» «software» dev1
«block» «software» DropAgent
«block» «software» Init1_Agent
«block» «hardware» push_motor Motor which controls push arm
«block» «hardware» press_motor motor which controls press cylinder
«block» «hardware» limit_switch switch which positions the press motor in right position
«block» «hardware» eject_motor Motor which controls eject arm
«block» «hardware» shredder_motor Motor which controls shredding and washing operations
«block» «hardware» ultrasonic_sensor Sensor which checks distance for anomaly and emergency situations
«block» «hardware» conveyor_motor_2 Conveyor belt motor which carries the product sort or build process
«block» «hardware» conveyor_motor_1 Conveyor belt motor which carries product on shredding and washing processes to sorting process
«block» «hardware» color_sensor Sensor which decides color of product
«block» «hardware» drop_sensor The sensor checks product before drop to conveyor belt
«block» «hardware» button Start button for whole system
«block» «hardware» drop_motor Motor drops the product to conveyor belt
controls
contains
reads(thread1)
reads(thread2)
reads

Fig. 1. Block definition diagram of the system.

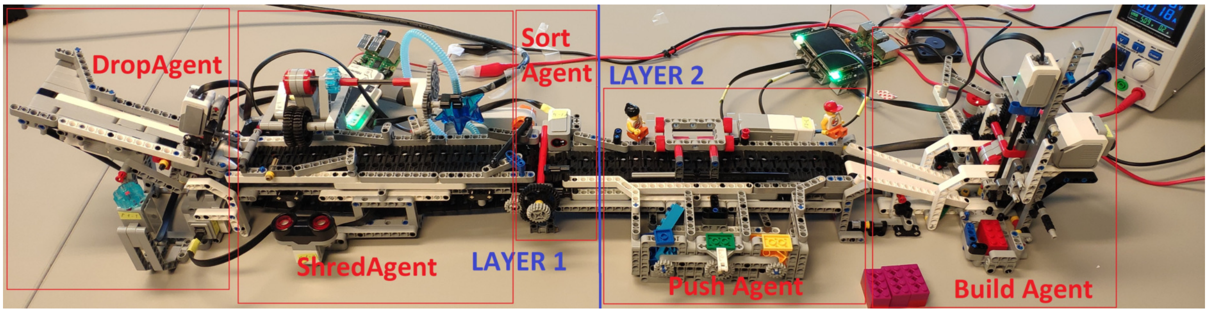


Fig. 2. Layers and agents of the Lego-based production System.

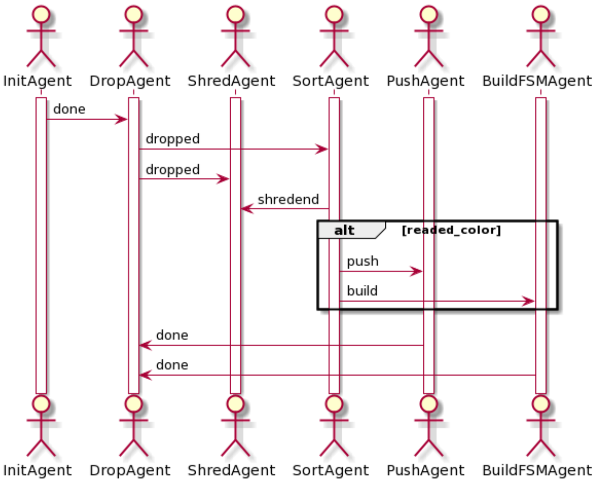


Fig. 3. Message sequence of the system agents.

1) Layer 1 Agents and their Behaviors:

Initialize Agent: Unpredictable power cuts and instant system shutdowns may cause positioning problems for the motors. When the power is cut, motors freeze at a position that is unknown by the system. Unknown motor positions cause failures on tasks. The main task of this agent is positioning the motors within mechanical limits. After motors are positioned, the agent sends a "done" message to the *Drop Agent*. This agent has a one-shot behaviour that works only once a time when the system starts up. There are 2 initializing agents for each layer. The initialize agent in *layer1* positions the drop motor with the mechanical limiter.

Drop Agent: Drop Agent is responsible for delivering the product (Lego brick) from system input to conveyor belt. It has a cyclic behaviour so it continuously samples data from 2 sensors while controls a motor. It waits for a "done" message from *Initialize Agent* or *Build Agent*, then the user presses the button to run the system continuously. The "done" message refers to the system is ready for the first run or the current process is done so that *Drop Agent* can deliver a new product to the conveyor belt.

Before *Drop Agent* runs the motor to drop a brick on the conveyor belt, it checks whether there is any brick in the input using the sensor at the input. If this condition is satisfied, then *Drop Agent* delivers the brick to the conveyor belt. It rotates the motor 90° clockwise to release the brick and then -90° anti-clockwise to return its initial position. Lastly, *Drop Agent* sends "dropped" message to *Shredder Agent* and *Sort Agent* to inform these agents about completion of its operation.

Shredder Agent: Shredder Agent is responsible for controlling shredding and washing processes. This agent has a continuous cyclic behaviour. The behaviour starts with receiving a "dropped" message from *Drop Agent* and stops when a "shredend" message is received from *Sort Agent*. While product shredding, washing and moving to the second conveyor belt, the agent concurrently checks an ultrasonic sensor with a thread. In case of any outside intervention, the system accepts this intervention as an emergency and stops the shredder motor, washing motor and conveyor belt.

Sort Agent: As represented in Figure 7, *Sort Agent* has major role for making decisions in the system. It executes a *Cyclic behavior*. After a product is dropped on the conveyor belt, *Sort Agent* starts waiting for a brick and activates the colour sensor. When the sensor realizes that the brick has arrived, it stops the conveyor belt. If the brick still does not arrive at the sensor after a certain time, the sort agent reverses the movement of the conveyor belt to set free the brick which is stuck. It reads colour sensor to recognize colour of the brick. Sensor sampling starts with receiving a "dropped" message and ends with product recognition. If the sensor recognizes product arrival to the sensor, then *Sort Agent* reads the colour of the brick and stops the conveyor belt. Then, it has 4 decision options to deliver brick and to inform related agents:

- Move brick to the bucket 1 and send "push" message to Push agent.
- Move brick to the bucket 2 and send "push" message to Push agent.
- Move brick to the bucket 3 and send "push" message to Push agent.
- Move brick to the press and send "build" message to Build agent.

2) Layer 2 Agents and their Behaviors:

Init2 Agent: Initializes the push motor, press motor and eject motor to their initial positions. This agent executes a one-shot

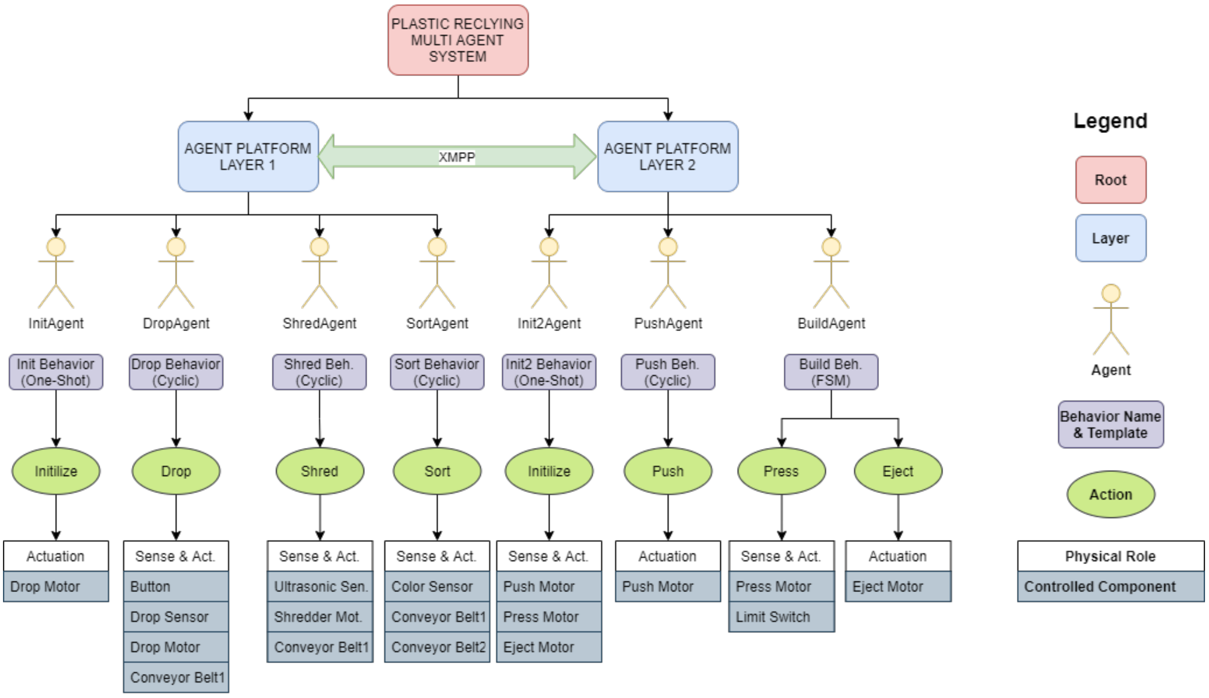


Fig. 4. Organisation diagram of the system.

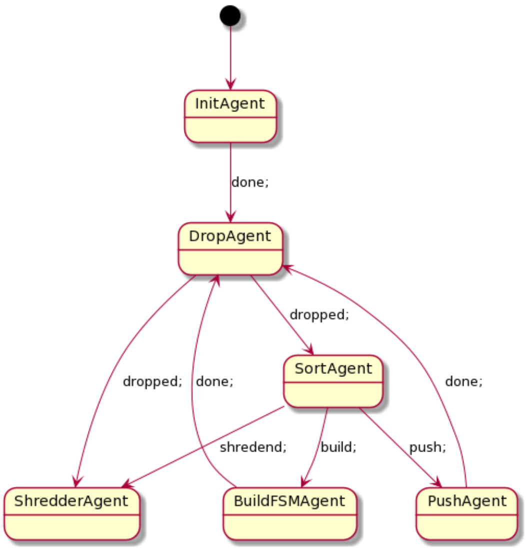


Fig. 5. Process transition between the agents.

behaviour. The agent is created with the system start up and dies after completing its behaviour and related task execution. In the Lego systems, the moving parts are usually limited with mechanical bounds. Therefore, we added an extra limit switch into this configuration for the press motor to obtain a much better initial position performance.

Push Agent: Push agent is an agent that has a cyclic behaviour. After it receives a "push" message, it pushes the mechanical line forward and then back. After, it sends the "done" message to *Drop agent* to inform the process is completed. When it receives a "push" message, *Push Agent* turns the motor clockwise with 120° and after a second, it turns counter-clockwise with 120°.

Build Agent: As Figure 6 illustrates, *Build Agent* controls the pressing process in an FSM manner. The agent starts pressing the first product after it receives the first "build" message. Then, it waits for the second "build" message which means the second product is about to arrive. When the agent receives the first "build" message, then it moves to *Press 1* state where it holds the first brick. Once it receives the second "build" message, then it switches to the *Press 2* state to combine these two bricks. After the completion of these two consecutive actions, *Build Agent* ejects the arm and pushes the products to the storage area. *Build Agent* executes its FSM behaviour continuously until the system shutdowns.

IV. Implementation of the Production Line

The software agents work in collaboration to control the heterogeneous parts of our production line which is, in fact, a complex CPS. These agents periodically sense their environment and operate to achieve their goals while keeping the system operational. Agents are self-containing entities that are able to achieve their tasks by providing local control for the different parts of the system. The role distribution to the agents are defined according to process phases to harvest the product and they are programmed to work in harmony with other agents. Their modularity and dynamic deployment also enhance the physical upgrades and changes, in other words,

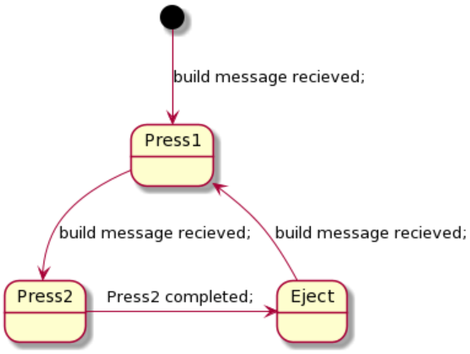


Fig. 6. State diagram of the Build agent.

new agents and new hardware can be added to the system easily.

In this section, the implementation of our smart manufacturing system is elaborated including the hardware setup and software agent implementation. The system configuration, the realization of the communication between agents, and the implementation of the corresponding behaviour classes are all discussed in the following subsections. The final structure of the implemented system has been previously shown in Figure 2.

A. System Configuration

The system is controlled by two *PiStorms* interface boards and two Raspberry Pi 3. Raspbian operating system runs Python 3.7 to interpret both embedded software and agent codes to control the system. SPADE is used for creating agents while PiStorms API is used to control Lego EV3 sensors/actuators.

In addition to the Lego production line pack, some modifications were made to resemble a more realistic industrial system. In the original system, the whole conveyor band had been controlled with only a single motor. Thanks to the modularity of Lego Technology, we separated conveyor bands to make each motor controls a separate conveyor belt so that two conveyor bands became controlled by the individual motors.

Generally, in most industrial production process implementations, limit switches are one of the most necessary hardware components to increase the reliability of the system. Hence, we added a limit switch for reducing the re-positioning error of the pressing process to zero shift. In case of any unexpected power cuts or environmental uncertainty, the system can obtain the initial position accurately using the limit switches.

Moreover, the initial version of the system had some issues about sampling colour value at the intersection point of *conveyor 1* and *conveyor2*. Sometimes there was some noise that effecting colour sampling data due to the moving parts. To fix this, we separated conveyors to find the optimal position for the colour sensor. Lastly, we added some brick parts as limiters to keep the moving bricks on the middle of the conveyors accurately.

B. Embedded Software

As discussed previously, we applied the singleton design pattern to restrict object creation from the class, including the hardware-specific I/O operations. Because the agents should access the same memory address and register so that an agent does not override other agent's access.

Inside the device-specific classes namely *dev1* and *dev2*, we also created inner classes for each hardware component. Inside these inner classes, there are functions specialized for each hardware element. For instance, an excerpt from the *ConveyorMotor* inner classes is given in Listing 1.

These inner classes can be accessed by an agent to control the hardware. Inside these inner classes, we developed a wrapper to raise the abstraction between the embedded Lego library and class implementation. In this way, wrapped code became more suitable for behavioural programming. In Listing 1, *start()*, *startSlow()*, *stop()*, *brickStucked()* and *runDegs()* functions are shown. These functions access the device-specific functions defined in the PiStorms library and wrap them to make them more usable for agent-based programming. Between lines 2 and 3, the conveyor motor is initialized and set to a certain speed. Lines 5 and 6 describe a lower speed setting for the conveyor motor while lines 8 and 9 instruct the stop function. When the system detects a stuck on the conveyor belt, it calls *brickStucked* function to reverse the conveyor belt. Lastly, lines between 14 and 16 define the *runDegs* method to rotate and run the motor according to the desired parameters.

Listing 1. ConveyorMotor inner class

```
1  class ConveyorMotor:
2      def start(self):
3          dev1.psm.BBM2.setSpeed(-100)
4          print(f'Conveyor Started')
5      def startSlow(self):
6          dev1.psm.BBM2.setSpeed(-20)
7          print(f'Conveyor Slow Started')
8      def stop(self):
9          dev1.psm.BBM2.setSpeed(0)
10         print(f'Conveyor Stopped')
11     def brickStucked(self):
12         dev1.psm.BBM2.runDegs(200, 100, True, False)
13         print(f'Brick stucked')
14     def runDegs(self,degree,speed):
15         dev1.psm.BBM2.runDegs(degree, speed, True, False)
16         motorState = dev1.psm.BBM2.isBusy()
17         print(f'Motor rotated {degree} degree on {speed}
              speed')
18         return motorState
```

It is the working principle of an agent to operate independently using behaviours and execute them in parallel with other agents. However, considering our I/O blocking situation, it is now possible to en-queue any sensor reading or motor actuating behaviours. Therefore, we need a concurrent system where it can run continuously without any interruption. Moreover, the system should sample data from the sensors while actuating a motor for 3 seconds in parallel with running another motor for 5 seconds.

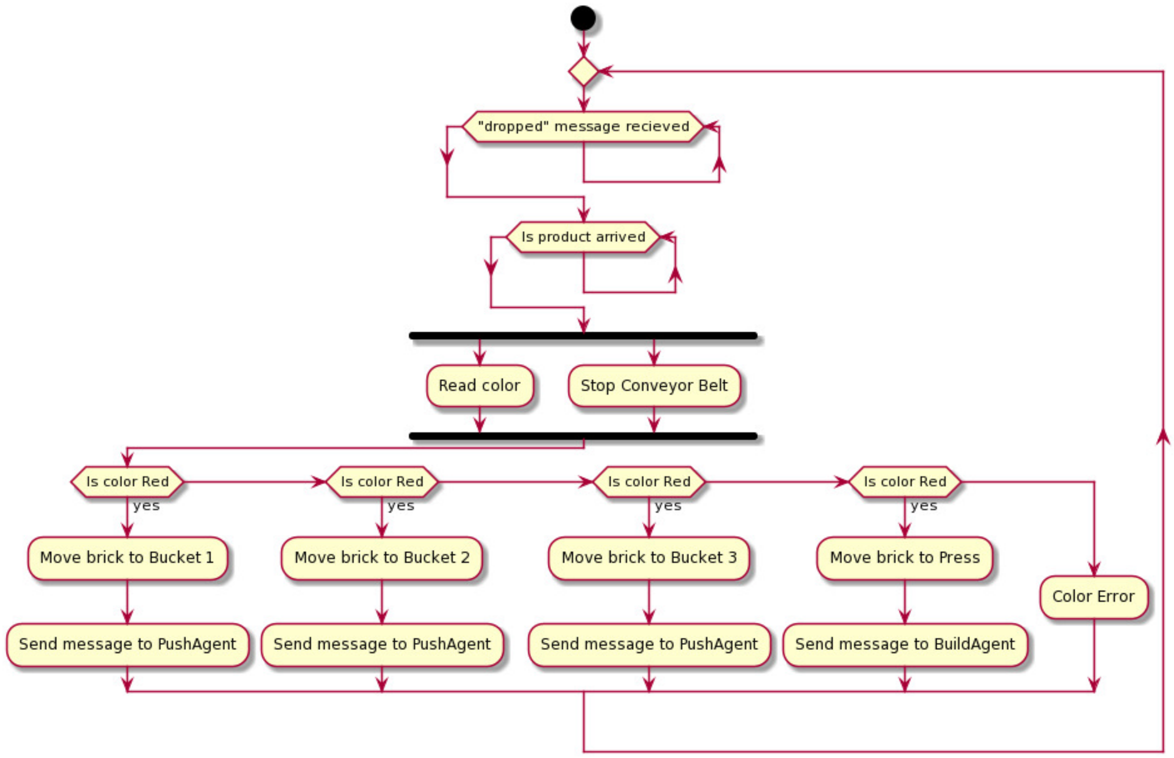


Fig. 7. Activity diagram of the Sort agent.

The obvious way to implement this concurrency is to assign a Python thread to each agent. However, there is a need for more parallelism within each agent, because an agent may also be involved in negotiations with other agents and each negotiation should proceed at its own pace while I/O blocking situations exist. In the implemented system, we used traditional threads for reading sensors and actuating motors instead of applying agent behaviours directly. Because sensor sampling is a crucial and continuous activity and agent behaviours can be blocked due to these I/O operations according to their processes considering the sampling rate. This considerably reduces the runtime-slices of each agent behaviour by blocking other operations when an agent reads the sensor inside these behaviours. As a solution, our implementation made each agent starts another thread within its setup and handles I/O operations.

Listing 2. Threading mechanism to avoid blocking I/O operations

```
1  threading .Thread( target =dev1.ColorSensor. waitBrick ,  args
       =(dev1 .ColorSensor ,) )
2  async  def  setup ( self ) :
3          print ("SortAgent  ::   started ")
4          b =  self .SortBeh()
5          template  = Template()
6          template . set_metadata (" performative " ,  "inform")
7          self .add_behaviour(b,  template )
8          print ("SortAgent  ::  running")
9          t . start  ()
```

In Listing 2, an excerpt from one of the created threads for the colour sensor is given. In line 1, the target method is defined. Lines between 2 and 9 describe the setup function of the agent which is also used for the initialization of the interrelated threads.

In Listing 3, a code excerpt from the LimitSwitch class which defines a limit switch is given. In the production line system, the limit switch is used to set the borders of motion of the components. For this purpose, *LimitSwitch* inner class is specialized for the limit switch hardware. In line 3, the state of the button is checked periodically, then "if/else" statement controls the state of the button. In this way, agents can detect the limit of motion when *isPressed()* function returns true and then they behave accordingly.

Listing 3. LimitSwitch innerclass

```
1  class  LimitSwitch:
2      def  isPressed ( self ) :
3          touch  = dev2.psm.BBS1.isTouchedEV3()
4          if  touch  != True:
5              return  False
6          else :
7              print (f'LS Pressed')
8              return  True
```

To minimize work accidents, many sensors are added to the manufacturing systems for occupational safety and health. These sensors must sense quickly as expected. For better sensor sampling rates and reactions, two threads of execution were implemented in the responsible agents: Shredder Agent executes the thread to check emergency while Sort Agent executes the thread on checking the arrival of bricks to the colour sensor.

Colour sensors can be influenced negatively by the noises in the environment. To remove this effect, we implemented a sensor sampler inside the system. The system collects data from samples arriving from the sensor. If the last 15 samples are the same, the system accepts the colour. Otherwise, it continues to sample data (see. Listing 4).

Listing 4. Color Sensor Sampling

```
def waitBrick( self ):
    readedcolorlist = [0]*15
    index = 0
    readSensor = True
    count = 0
    print (" Starting to wait brick")
    dev1. retVal = 0.0
    while readSensor:
        color = dev1.psm.BBS2.colorSensorEV3()
        readedcolorlist [index] = color
        now = datetime.now()
        index = index + 1
        x=sum( readedcolorlist )/15
        if index ==15:
            index=0
            if x==2 or x==3 or x==4 or x==5 or x==6:
                print ( str (index) + " −> " + "
                        ReadedColor:" + str ( color ))
                print ("RETVAL:",str(x))
                dev1. retVal = x
            else :
                dev1. retVal = 0.0
        time . sleep (0)
```

Sample videos demonstrating how the implemented system executes the continuous production and manages a stuck event in the production line are available at https://youtu.be/dRUyXYuDPlY and https://youtu.be/_xgYyaBMv90.

V. Discussion

CPPS are expected to provide various features such as adaptiveness, awareness, intelligence, and abstraction to meet the requirements of the emerging industrial applications. Agent-based approaches can be a good alternative to support these features. However, an integration of the industrial systems with the agents is still a significant issue for the agentification of such systems as discussed in [33] and [36]. MAS is a paradigm derived from the distributed artificial intelligence field that covers distribution, decentralization, intelligence, autonomy and adaptation. Using these features, MAS provide flexibility, robustness, responsiveness and reconfigurability and create an ecosystem of intelligent, autonomous and cooperative computational entities. Despite the fact that MAS technology has already been integrated into several industrial applications such as smart production, smart power grids, smart logistics and smart healthcare, acceptance and standardisation of industrial agents is still under debate.

Seamless integration of MAS, embedded system and CPS may bring solutions to the abovementioned issues and lead to the realization of the expected features. Since CPS consists of both physical and cyber parts, top-level methodologies cannot be evaluated and shown without low-level architectures to emulate the industrial problems. There is no uniform way to integrate the software agents to the low-level automation functions to utilize them as the industrial agents [32]. Hence, the miniaturisation of the industrial systems, mimicking the process steps and reproducing the problems as described in our study can be a way to ease the burden of developing industrial agents within this context.

To achieve CPS and agent integration, device specific libraries are mostly required. Then, these libraries can be merged with agent development environments. The library can be wrapped to provide behavioral structures. Once the control of the physical components is achieved by the cyber side, the agentification process can be applied. Moreover, the integration process can be facilitated by using software engineering design principles e.g. benefiting from the design patterns. Moreover, the physical construction of the target system is still required because CPS is also a physical entity. To address this requirement, we suggested using the Lego technology, which allows the miniaturisation of the interaction between embedded systems and agents while providing extensibility for applying high-level solutions and mechanisms. When the MAS is integrated into any system, the agents inside can be distributed to the subsystems to achieve the control process distribution while establishing a network for negotiation and messaging. In this way, functionalities of the embedded devices can be encapsulated into the behaviours of the agents. Then, various behaviours can be defined for executing tasks, sending parameters, and controlling the process to achieve the system goals.

After the completion of the agentification process, the system can also be enhanced with the distributed wireless sensors for data acquisition [37], [38]. The edge, fog and cloud computing can be the key enabler technologies for CPPS considering IoT and CPS interoperability. Then, this data can be fed into the Machine Learning algorithms to achieve various computations such as pattern matching to detect system faults, prediction algorithms to avoid human errors and system-level reasoning to apply high-level plans.

During the implementation of the smart production line introduced in this paper, we followed some fundamental industrial application principles. Firstly, to keep the pressing operation calibrated, the limit switch was added to measure the elevation. While the press goes up and down, it touches the limit switch so that it operates between bounded limits. Secondly, we followed the separation of concerns principle and placed an agent for a section of the production line. In other words, only one agent is responsible for a process phase. Lastly, the same principle was also applied to conveyor belts to create *layer 1* and *layer 2*. When the task is finished in the *layer 1*, it delivers the product to the *layer 2* so that *layer 1* can receive a new task while *layer 2* processing the second step of the previous task operating as pipelined.

We believe that the constructed system based on the Lego technology may be an appropriate tool for education considering the CPS and agent integration. Due to the fact that CPS is a multi-disciplinary field and owns multi-target domains, it is studied by a lot of researchers, engineers and practitioners. However, the recent advancements, open issues and challenges require multi-disciplinary knowledge as CPS has a wide umbrella that unites various engineering fields and disciplines. Most of the engineering and information technology courses now focus on CPS, agent-based programming and embedded technology and the requirement of autonomy and intelligence mostly becomes a must to achieve and sustain next-generation systems [39]. We need physically easy-to-construct and easy-to-modify technologies integrated with easy-to-deploy and easy-to-run programming paradigms. Lego technology provides modular and modifiable structures to meet these requirements while agent-oriented approaches present higher-level abstraction of programming. Moreover, the nature of the agents paves the way for integrating artificial intelligence, inter-operating IoT solutions, and high-level programming. As a result, multi-disciplinary studies can be taught to the future's talented engineers and students using our proposed approach.

As some technical notes, we would like to share that we faced with some challenges during the operation of the system. Due to the power requirement of RaspberryPI, Lego components and PI Storms, the system was fed with two power supplies and each power supply was feeding the system with 9.8 Volts and 3 amps. Alternatively, Li-Po batteries can also be used for short-term tests and mobility. Because the power requirements cannot be fed, then the motors fail, and the system shuts down. Moreover, if the motors get heated, then cold gels of the spray should be applied to cool down the components. To reduce the friction between Lego bricks and moving parts, we used machine oil.

Lastly, during the sensor sampling, we discovered that the colour sensor could not recognize the colour of the Lego bricks accurately due to the speed of the conveyor belt. Instead of reducing the velocity of the conveyor belt, we provided a buffered reading at the cyber part by wrapping the method into the sensor reading method and physical buffered transition by moving the colour sensor between two conveyor belts. Naturally, when Lego bricks are transferred from the first conveyor belt to the second one (*layer 1* to *layer 2*), we benefited from the natural delay caused by the friction between them. This delay and buffered reading raised accurate decisions on the colour of the Lego bricks. This decision can be supported by using pattern matching algorithms, machine learning, and/or dynamic buffer size. Because we are aware that selecting industrially standardised sensors does not guarantee ideal operation and reducing the sensor errors under harsh and corrosive conditions is another challenge [40].

VI. Conclusion

In this paper, a system to integrate software agents and CPS is proposed based on SPADE, RaspberryPI and Lego technologies. The design and implementation of this production line system are discussed. With employing agents and encapsulating embedded functions, an agent-based control on the CPPS is achieved. In this way, it is also avoided to deal with low-level details of embedded software for robot programming. Also, the distributed and mobility capabilities of software agents helped to develop heterogeneous components in the system. Our system based on the Lego technology may also assist the education activities especially considering how automation on CPPS can be supported via software agents.

As a future study, we aim to improve the current reasoning and planning capabilities of the agents in our system using belief-desire-intention (BDI) logic [41]. Additionally, we intend to provide a multi-paradigm approach, e.g. by benefiting from the IoT paradigm, so that our system both works with the same system instances (homogeneous infrastructures) and incorporate with different type systems (heterogeneous infrastructures) by establishing a network. For this purpose, both the state-of-the-art on agent-based IoT systems, as well as our past experiences, [42] will be considered. In addition, a model-based framework can also be developed to support the current development process by automatically synthesizing both agent code and embedded software [43]. To achieve this, model-driven engineering techniques similar to the ones we used in [44], [45] can be applied again to these systems to reduce the complexity.

References

[1] R. Baheti and H. Gill, "Cyber-physical systems," *The impact of control technology*, vol. 12, no. 1, pp. 161–166, 2011, doi: https://doi.org/10.1109/icmech.2019.8722929.

[2] L. Monostori, B. Kádár, T. Bauernhansl, S. Kondoh, S. Kumara, G. Reinhart, O. Sauer, G. Schuh, W. Sihn, and K. Ueda, "Cyber-physical systems in manufacturing," *Cirp Annals*, vol. 65, no. 2, pp. 621–641, 2016, doi: https://doi.org/10.1016/j.cirp.2016.06.005.

[3] K.-D. Thoben, S. Wiesner, and T. Wuest, ""industrie 4.0" and smart manufacturing-a review of research issues and application examples," *International journal of automation technology*, vol. 11, no. 1, pp. 4–16, 2017, doi: https://doi.org/10.20965/ijat.2017.p0004.

[4] N.-H. Tran, H.-S. Park, Q.-V. Nguyen, and T.-D. Hoang, "Development of a smart cyber-physical manufacturing system in the industry 4.0 context," *Applied Sciences*, vol. 9, no. 16, p. 3325, 2019, doi: https://doi.org/10.3390/app9163325.

[5] M. Challenger, B. T. Tezel, V. Amaral, M. Goulao, and G. Kardas, "Agent-based cyber-physical system development with sea_ml++," in *Multi-Paradigm Modelling Approaches for Cyber-Physical Systems*, B. Tekinerdogan, V. Amaral, and H. Vangheluwe, Eds. Elsevier Pub., 2021, doi: https://doi.org/10.1016/B978-0-12-819105-7.00013-1.

[6] T. Semwal, M. Bode, V. Singh, S. S. Jha, and S. B. Nair, "Tartarus: a multi-agent platform for integrating cyber-physical systems and robots," in *Proceedings of the 2015 Conference on Advances in Robotics*, 2015, pp. 1–6.

[7] P. Leitao, S. Karnouskos, L. Ribeiro, J. Lee, T. Strasser, and A. W. Colombo, "Smart agents in industrial cyber–physical systems," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1086–1101, 2016, doi: https://doi.org/10.1109/JPROC.2016.2521931.

[8] E. Schoofs, J. Kisaakye, B. Karaduman, and M. Challenger, "Software agent-based multi-robot development: A case study," in *2021 10th Mediterranean Conference on Embedded Computing (MECO)*. IEEE, 2021, pp. 1–8, doi: https://doi.org/10.1109/MECO52532.2021.9460210.

[9] B. Vogel-Heuser, J. Lee, and P. Leitão, "Agents enabling cyber-physical production systems," *at-Automatisierungstechnik*, vol. 63, no. 10, pp. 777–789, 2015, doi: https://doi.org/10.1515/auto-2014-1153.

[10] E. Negri, L. Fumagalli, and M. Macchi, "A review of the roles of digital twin in cps-based production systems," *Procedia Manufacturing*, vol. 11, pp. 939–948, 2017, doi: https://doi.org/10.1016/j.promfg.2017.07.198.

[11] J. Ding, Z. Li, and T. Pan, "Control system teaching and experiment using lego mindstorms nxt robot," *International Journal of Information and Education Technology*, vol. 7, no. 4, p. 309, 2017.

[12] D. Gauntlett, "The lego system as a tool for thinking, creativity, and changing the world," *Lego studies: Examining the building blocks of a transmedial phenomenon*, pp. 1–16, 2014, doi: https://doi.org/10.4324/9781315858012.

[13] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman *et al.*, "Scratch: programming for all," *Communications of the ACM*, vol. 52, no. 11, pp. 60–67, 2009.

[14] F. Erata, M. Challenger, B. Tekinerdogan, A. Monceaux, E. Tüzün, and G. Kardas, "Tarski: A platform for automated analysis of dynamically configurable traceability semantics," in *Proceedings of the 32nd ACM SIGAPP Symposium on Applied Computing*, 2017, pp. 1607–1614, doi: https://doi.org/10.1145/3019612.3019747.

[15] J. Holt and S. Perry, *SysML for systems engineering*. IET, 2008, vol. 7.

[16] M. E. Gregori, J. P. Cámara, and G. A. Bada, "A jabber-based multi-agent system platform," in *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, 2006, pp. 1282–1284.

[17] L. P. GitHub, "Lego PiStorms Lubrary," Available:{https://github.com/mindsensors/PiStorms}, [Online; accessed 9-May-2021].

[18] S. Demirkol, S. Getir, M. Challenger, and G. Kardas, "Development of an agent based e-barter system," in *2011 International Symposium on Innovations in Intelligent Systems and Applications*. IEEE, 2011, pp. 193–198, doi: https://doi.org/10.1109/INISTA.2011.5946060.

[19] M. Merdan, M. Vallee, W. Lepuschitz, and A. Zoitl, "Monitoring and diagnostics of industrial systems using automation agents," *International journal of production research*, vol. 49, no. 5, pp. 1497–1509, 2011, doi: https://doi.org/10.1080/00207543.2010.526368.

[20] V. Mascardi, D. Weyns, A. Ricci, C. B. Earle, A. Casals, M. Challenger, A. Chopra, A. Ciortea, L. A. Dennis, Á. F. Díaz *et al.*, "Engineering multi-agent systems: State of affairs and the road ahead," *ACM SIGSOFT Software Engineering Notes*, vol. 44, no. 1, pp. 18–28, 2019, doi: https://doi.org/10.1145/3310013.3322175.

[21] S. Jeschke, C. Brecher, T. Meisen, D. Özdemir, and T. Eschert, "Industrial internet of things and cyber manufacturing systems," in *Industrial internet of things*. Springer, 2017, pp. 3–19, doi: https://doi.org/10.1007/978-3-319-42559-7_1.

[22] N. Jazdi, "Cyber physical systems in the context of industry 4.0," in *2014 IEEE international conference on automation, quality and testing, robotics*. IEEE, 2014, pp. 1–4.

[23] E. A. Lee, "Cyber physical systems: Design challenges," in *2008 11th IEEE international symposium on object and component-oriented real-time distributed computing (ISORC)*. IEEE, 2008, pp. 363–369, doi: https://doi.org/10.1109/ISORC.2008.25.

[24] P. Leitão, J. Barbosa, C. A. Geraldes, and J. P. Coelho, "Multi-agent system architecture for zero defect multi-stage manufacturing," in *Service Orientation in Holonic and Multi-Agent Manufacturing*. Springer, 2018, pp. 13–26, doi: https://doi.org/10.1007/978-3-319-73751-5_2.

[25] W. Xing, Y. Jun, L. Peihuang, and T. Dunbing, "Agent-oriented embedded control system design and development of a vision-based automated guided vehicle," *International Journal of Advanced Robotic Systems*, vol. 9, no. 2, p. 37, 2012.

[26] J. Queiroz, P. Leitão, J. Barbosa, and E. Oliveira, "Distributing intelligence among cloud, fog and edge in industrial cyber-physical systems," in *16th International Conference on Informatics in Control, Automation and Robotics, ICINCO 2019*, 2019, pp. 447–454.

[27] I. Horváth, Z. Rusák, and Y. Li, "Order beyond chaos: Introducing the notion of generation to characterize the continuously evolving implementations of cyber-physical systems," in *ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers Digital Collection, 2017, doi: https://doi.org/10.1115/DETC2017-67082.

[28] A. Petrovska, M. Neuss, I. Gerostathopoulos, and A. Pretschner, "Run-time reasoning from uncertain observations with subjective logic in multi-agent self-adaptive cyber-physical systems," in *16th Symposium on Software Engineering for Adaptive and Self-Managing Systems, SEAMS*, 2021, doi: https://doi.org/10.1109/SEAMS51251.2021.00026.

[29] G. Kardas, Z. Demirezen, and M. Challenger, "Towards a dsml for semantic web enabled multi-agent systems," in *Proceedings of the International Workshop on Formalization of Modeling Languages*, ser. FML '10. New York, NY, USA: Association for Computing Machinery, 2010. [Online]. Available: https://doi.org/10.1145/1943397.1943402

[30] M. Challenger, B. T. Tezel, O. F. Alaca, B. Tekinerdogan, and G. Kardas, "Development of semantic web-enabled bdi multi-agent systems using sea_ml: An electronic bartering case study," *Applied Sciences*, vol. 8, no. 5, 2018, doi: https://doi.org/10.3390/app8050688. [Online]. Available: https://www.mdpi.com/2076-3417/8/5/688

[31] T. Semwal and S. B. Nair, "Agpi: Agents on raspberry pi," *Electronics*, vol. 5, no. 4, p. 72, 2016.

[32] P. Leitão, S. Karnouskos, L. Ribeiro, P. Moutis, J. Barbosa, and T. I. Strasser, "Common practices for integrating industrial agents and low level automation functions," in *IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2017, pp. 6665–6670, doi: https://doi.org/10.1109/IECON.2017.8217164.

[33] S. Karnouskos, P. Leitao, L. Ribeiro, and A. W. Colombo, "Industrial agents as a key enabler for realizing industrial cyber-physical systems: Multiagent systems entering industry 4.0," *IEEE Industrial Electronics Magazine*, vol. 14, no. 3, pp. 18–32, 2020, doi: https://doi.org/10.1109/MIE.2019.2962225.

[34] I. O. for Standardization, *Systems and Software Engineering: Systems and Software Quality Requirements and Evaluation (SQuaRE): Measurement of System and Software Product Quality*. ISO, 2016.

[35] A. Hornsby and R. Walsh, "From instant messaging to cloud computing, an xmpp review," in *IEEE International Symposium on Consumer Electronics (ISCE 2010)*. IEEE, 2010, pp. 1–6.

[36] L. Sakurada and P. Leitão, "Multi-agent systems to implement industry 4.0 components," in *2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS)*, vol. 1. IEEE, 2020, pp. 21–26, doi: https://doi.org/10.1109/ICPS48405.2020.9274745.

[37] B. Karaduman, T. Aşıcı, M. Challenger, and R. Eslampanah, "A cloud and contiki based fire detection system using multi-hop wireless sensor networks," in *Proceedings of the Fourth International Conference on Engineering & MIS 2018*, 2018, pp. 1–5, doi: https://doi.org/10.1145/3234698.3234764.

[38] B. Karaduman, M. Challenger, and R. Eslampanah, "Contikios based library fire detection system," in *2018 5th International Conference on Electrical and Electronic Engineering (ICEEE)*, 2018, pp. 247–251, doi: https://doi.org/10.1109/ICEEE2.2018.8391340.

[39] J. Tavčar and I. Horváth, "A review of the principles of designing smart cyber-physical systems for run-time adaptation: Learned lessons and open issues," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 1, pp. 145–158, 2018, doi: https://doi.org/10.1109/TSMC.2018.2814539.

[40] K. Thiyagarajan, S. Kodagoda, L. Van Nguyen, and R. Ranasinghe, "Sensor failure detection and faulty data accommodation approach for instrumented wastewater infrastructures," *IEEE Access*, vol. 6, pp. 56 562–56 574, 2018, doi: https://doi.org/10.1109/ACCESS.2018.2872506.

[41] B. T. Tezel, M. Challenger, and G. Kardas, "A metamodel for jason bdi agents," in *5th Symposium on Languages, Applications and Technologies (SLATE'16)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016, doi: https://doi.org/10.4230/OASIcs.SLATE.2016.8.

[42] N. Karimpour, B. Karaduman, A. Ural, M. Challenger, and O. Dagdeviren, "Iot based hand hygiene compliance monitoring," in *2019 International Symposium on Networks, Computers and Communications (ISNCC)*. IEEE, 2019, pp. 1–6, doi: https://doi.org/10.1109/ISNCC.2019.8909151.

[43] M. Challenger and H. Vangheluwe, "Towards employing abm and mas integrated with mbse for the lifecycle of scpsos," in *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*, 2020, pp. 1–7, doi: https://doi.org/10.1145/3417990.3421439.

[44] B. Karaduman, M. Challenger, R. Eslampanah, J. Denil, and H. Vangheluwe, "Platform-specific modeling for riot based iot systems," in *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*, 2020, pp. 639–646, doi: https://doi.org/10.1145/3387940.3392194.

[45] T. Z. Asici, B. Karaduman, R. Eslampanah, M. Challenger, J. Denil, and H. Vangheluwe, "Applying model driven engineering techniques to the development of contiki-based iot systems," in *2019 IEEE/ACM 1st International Workshop on Software Engineering Research & Practices for the Internet of Things (SERP4IoT)*. IEEE, 2019, pp. 25–32, doi: https://doi.org/10.1109/SERP4IoT.2019.00012.

Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 533–536

DOI: 10.15439/2021F17
ISSN 2300-5963 ACSIS, Vol. 25

Agile Architecting of Distributed Systems for Flexible Industry 4.0

Henrik Bærbak Christensen
Computer Science
University of Aarhus
Aarhus, Denmark
Email: hbc@cs.au.dk

Sune Chung Jepsen, Torben Worm
Software Engineering
University of Southern Denmark
Odense, Denmark
Email: {sune, tow}@mmmi.sdu.dk

***Abstract*—Small and medium sized businesses within mechanical manufacturing cannot benefit from Industry 4.0 automation as small production batches are unable to pay for up-front robotic configuration and programming costs. In this paper, we report on early results from a project aiming at developing a software architecture supporting fast, easy, and flexible reconfiguration of a robotic manufacturing process, using an agile and prototyping approach.**

I. Introduction

ROBOTIC manufacturing is well adopted in for instance the automotive industry. Such manufacturing is characterized by production of large volumes of nearly identical products which can justify high cost of setting up the production line and programming robots in the production-line. However, small and medium-sized businesses (SMB) in mechanical production often have low production volumes, often just a single or less than 10 products. Thus, adopting robotic manufacturing, Industry 4.0—*the intelligent networking of machines and processes for industry with the help of information and communication technology* [7], is challenging for SMBs. In our project, we are exploring flexible production, customization, and handling changing requirements in collaboration with a number of Danish SMBs.

Our main contribution is early results from applying *architectural prototyping* to formulate distributed architectures for Industry 4.0 with an emphasis on flexibility as a central quality attribute. A second contribution is early architectural insights from this work, which adopts an agile and run-time focus in contrast to prevalent work that achieves flexibility through elaborate ontologies [8], [9] which in turn require intensive up-front engineering efforts [3].

II. Background

The project is a collaboration between three SMB within the mechanical production area (machine shops) as well as two Danish universities with competences within robotics and software architecture.

The main research challenge is:

> *Design a distributed systems software architecture that allows flexible production specification, adaptable to a small machine shop, providing high usability by workers trained in mechanical production.*

Ideally, a skilled worker (but with little prior computer science training) should be able to set up individual machine functions (metal cutting, shaping, drilling, assembling, packaging, etc.) as well as workflow (process order, movement of product between machines, etc.) fast and easy.

A schematic example is show in Figure 1 in which a worker (1) defines a workflow plan to be handled by the platform, which instructs transport robots (3 + 9), to move proper materials from a raw warehouse (2) to specified production cells/programmable robots (5 + 7) that do assembly, drilling, cutting, etc. Production cells may also receive/deliver materials using a magnetic track (4 + 6), before the final product is delivered to a finished goods warehouse (10).

III. Agile Architecture Design

Architecturally, a robotic machine shop is a *distributed system*, having independent nodes with specialized capabilities, adaptable by programming. The key challenge is designing a software architecture that supports flexible (re)configuration, ease of defining processes and workflow, as well as efficient/cost-effective production.

As a research project, another goal is to experiment with the architectural design space in a efficient manner in order to allow stakeholders early and agile feedback. *Architectural prototyping* [2], [4] matches these requirements, as it emphasizes architectural learning and exploring, using lightweight development of executable demonstration systems.

Architectural prototyping is an incremental and iterative process in which architectural design is postulated as a hypothesis, programmed in an architectural prototype (AP), and next validated for feasibility. Based upon the outcome, the AP is either rejected or refined, similar to a scientific process. This way the architectural design space is explored and refined, more than rigorously designed and evaluated up-front.

We therefore hypothesized a software architecture for a robotic machine shop, as exemplified in Figure 1, to be a *distributed system of programmable nodes (production cell, transports, warehouses, etc.) orchestrated by the software architecture pattern "Blackboard"* [1].

In the Blackboard pattern, individual nodes, *knowledge sources*, contribute data and events to the *blackboard* (i.e. a repository) while a *control plane* actively monitors the

IEEE Catalog Number: CFP2185N-ART ©2021, PTI

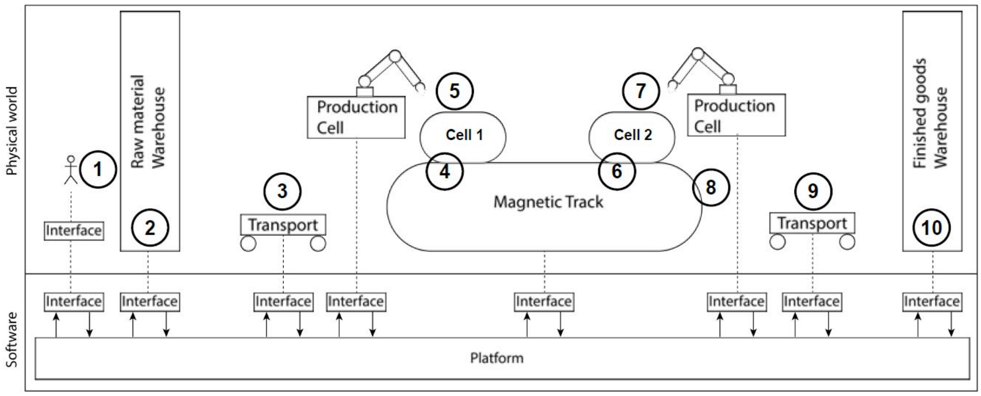


Fig. 1. Schematic Machine Shop [6]

TABLE I
DEVELOPED ARCHITECTURAL PROTOTYPES. SOURCE OF DATA IS A) DEVELOPER'S WORKING HOUR REGISTRATION, B) GIT LOG MESSAGE TIMESTAMPS, AND C) GIT BLAME LOGS OF DEVELOPMENT DIARIES FOR EACH AP.

AP No.	Goal	Outcome	Hour count
1	Establish Modelling and Blackboard Pattern	Accept	9h
2	Demonstrator for stakeholders	Accept	3.5h
3	Workflow. Insight: Carrier concept missing	Accept	7h
4	Knowledge Engine (JESS) Learning	Accept	3h
5	Carrier Introduced. JESS Integration	Accept/Failure	10.5h
6	EasyFlow Learning	Accept	1.5h
7	Workflow using EasyFlow. Demonstrator	Cond Accept	10h

blackboard, picking up state changes and issues new actions to be performed by nodes. To exemplify in a robotic machine shop context, a production cell may notify the blackboard that drilling a hole in a plate is finished, which trigger that the control plane tells the transport robot to move the finished plate to the assembly robot, etc.

Our first AP, see Table I, developed the core concepts adhering to the blackboard architecture, with a strong emphasis on *modeling physical objects and processes with computational equivalents* that support fast experimentation. Central examples of "computational equivalents", developed in our APs are:

- *Physical material*: Modelled by strings. Example: A bolt is just the string "B", a metal plate of 100x20x10 mm is just "P/100-20-10", etc.
- *Production cells*: Modelled by threads/processes, that receive materials from an in-queue, perform a "Production cell function" on the materials, before emitting it to its out-queue.
- *Production cell function*: Modelled by Strategy pattern, an algorithm to process material from one form to another. Example: Drilling a 3mm hole in the above plate "P/100-20-10" at position (10mm, 20mm) will return material "drill/3-10-20(P/100-20-10)". Note how the string just is a recursive specification of functions applied. To simulate time taken for a given function, delays are part of a cell function's execution.
- *Transport of Materials*: Also modelled by threads, but their "function" is set to move material from the out-queue of one production cell to the in-queue of the another. Essentially, just copying strings from an out-queue to an in-queue, with a delay.
- *Warehouse*: Again, just a thread, whose "function" is either to provide (raw warehouse) or store (finished goods) material. That is, it is just a collection of strings.
- *Queues (at cells)*: To simplify we used a blocking queue with just room for one material/string. Later changed to contain *Carrier* objects, see below.
- *Control plane*: Initially, we hard-coded the simplest possible production scenario, we could think of—one cell drills a hole in a plate, which is then moved to a second cell that screws a nut+bolt through the hole in the plate. That is, going from raw materials (P, B, N) to "screw(B, N, drill(P))" (dimensions omitted for clarity).

Executing an AP simply produces log messages from each thread outlining the actions taken by each transport ("MoveBot" in output below) or production cell ("Station" below), as exemplified in Figure 2.

Note the efficiency of the approach in exploring the control plane architectural aspects (Table 1): Only 9 hours was spent to establish an architectural sketch, defined core concepts, and their computational equivalents. A further 3½ hours was spent to polish the AP into a form that allowed demonstration to the project's SMB stakeholders.

```
[INFO] MoveBot :: MoveBot 1 - PICK UP material 'P/100-20-10'
[INFO] MoveBot :: MoveBot 1 - Start moving to Station 'Drill Station'
...
[INFO] MoveBot :: MoveBot 1 - DELIVER TO Drill Station
...
[INFO] Station :: Drilling 66%...
[INFO] Station :: Drilling 100% - producing 'drill/3-10-20(P/100-20-10)'
[INFO] Station ::  -- Adding to OUT QUEUE
[INFO] MoveBot :: MoveBot 1 - retrieved job:
       MoveJob{source='Drill Station', destination='Assembly Station'}
```

Fig. 2. AP-2 demonstrates workflow and actions performed through log messages (. . . indicates portions omitted for brevity)

The conclusion of AP-1 and AP-2 was that stakeholders judged that core modeling concepts were feasible, but "programming the control plane was tedious". The APs only supported a single workflow scenario, and involved lots of tedious and hand-coded handling of threads and queues. Another outcome of the AP-2 demonstration session was the need to introduce strong support for "state machines to model workflows".

APs 3–7, in Table 1, represent steps and sidesteps in exploring implementing a flexible, usable, control plane based upon state machines. AP-3's focus was on introducing simple statemachines, but quickly lead to the conclusion that a *Carrier* concept was missing:

- *Carrier*: A physical tray, organizing a set of materials in predefined positions for easy idenfication by a robot, ala "Pick the nut in position 3", see Figure 3. Our computational equivalents was an array, indexing a set of strings (representing materials), as well as the product's associated state machine.
- *Control plane*: Rewritten from the fixed workflow of AP-1 into a listener on any state change (production cells or transports finishing their tasks) which in turn leads to deciding on next state transition.

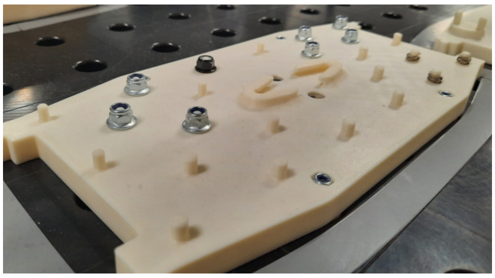


Fig. 3. A *carrier* from the Robotic Lab, holding nuts and springs.

The carrier concepts was introduced and validated in AP-5. Two APs were dedicated to exploring suitable libraries for programing state-machines (AP 4+6), before settling on EasyFlow for the stakeholder demonstrator. The (so far) final AP-7 was demonstrated at a second workshop to stakeholders—showing three workflows, producing two complex and one simple product.

One key outcome was that the carrier besides holding materials, also embody the state of the materials from its journey from a set of raw materials to the final product. Alas, the workflow statemachine is directly tied to the carrier. A second outcome was how the Blackboard architecture automatically optimizes the flow of material and processes in the production line: As soon as a production cell has finished, its carrier is available at the out-queue and the blackboard/control plane is notified. The control plane makes the state transition of the carrier's state machine, typically finding an idling transport robot to pick up the carrier; or selecting an available production cell that can serve the manufactoring task at hand: drilling, cutting, assembly, etc.

However, while our AP-7 validated the architectural design, the definition of workflows via state machines was still requiring low level programming, far from the required usability requirements.

Never-the-less, only a total of 44.5 hours was spent to establish a sound architectural basis validating an architectural approach based upon the blackboard pattern, carriers associated with product's state machines, and central concept implementations—as well as rejected several ideas, such as using knowledge engines, with little wasted effort. Future work focuses on bringing the state-machine programming into a user friendly format, likely exploring visual tools for generating state machines, like Visual Paradigm, SinelaboreRT, and of course proof of concept of the architecture by letting it control real manufacturing production cells and transports at the Robotic Lab at University of Southern Denmark.

A. Preliminary Architecture

In Figure 4, our preliminary architecture is sketched, using UML class diagram notation: Associations are annotated with essential behavior. Robot Units are general and represent independent processes such as the production cells, the warehouses, and transports like magnetic track or transport robots. The Blackboard controls them by upload programs, "functions", to them. Carriers contain the product as a set of (partially processed) materials, as well as the state machine representing the state of the partially processed product. Any state transitions (like a cell finishing and moving the carrier to its out queue) notify the Blackboard which determine next actions which is always "moving" carrier from one unit's out queue to the next unit's in queue. The actual sequence of units to visit and functions to apply is determined by the particular state machine of the product.

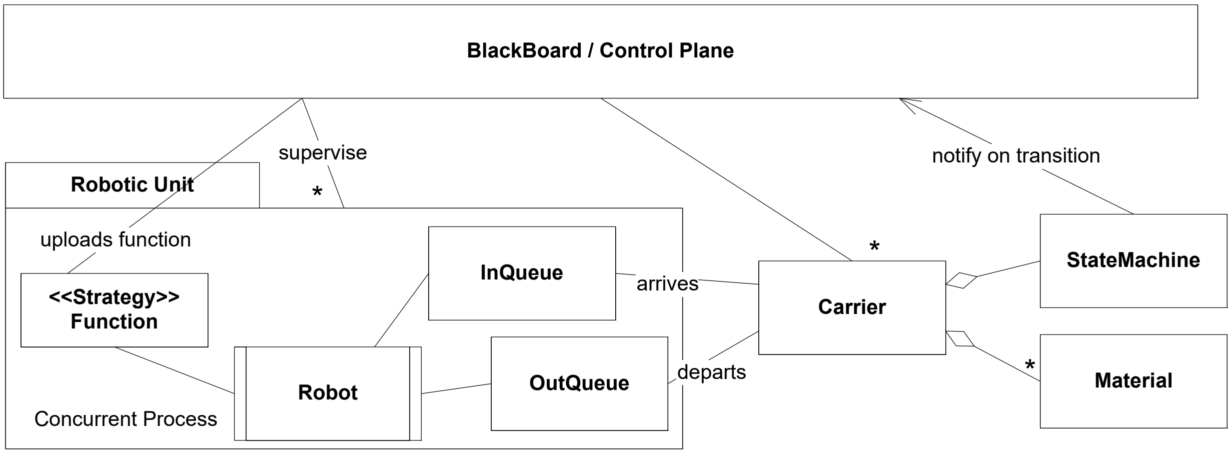


Fig. 4. UML Class diagram of preliminary architecture

We argue that even this preliminary architecture achieves a degree of flexibility, as the input for any given product is just the specification (set of functions, state sequence): that is, define the set of processings (drilling, assembly, cutting, etc.) as well as the ordering—drilling must be performed before assembling, which again must be performed before storing in the final warehouse, etc.

IV. Conclusion

In this paper, we have presented initial results on agile architecture development for distributed systems for SMB machine shops, with an emphasis on flexible and easy reconfiguration of manufacturing. We have presented results from early architectural design work based upon architectural prototyping, and shown how this technique provides fast feedback in the architectural work, allowing us to establish a solid architectural basis for further work in less than 45 hours of staff time. Furthermore, we have presented our suggestions on how the complex mechanical and distributed nature of a machine shop can be translated into computational equivalents that allows a fast and experimental development cycle in collaboration with workers in mechanical production.

Acknowledgements

The authors wants to acknowledge the financial support of InfinIT [5].

References

[1] Paris Avgeriou and Uwe Zdun. Architectural Patterns Revisited—A Pattern Language. In *Proceedings of 10th European Conference on Patterns Languages of Programming*, 2005.

[2] J. E. Bardram, H. B. Christensen, and K. M. Hansen. Architectural Prototyping: An Approach for Grounding Architectural Design and Learning. In *Proceedings. Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA 2004)*, pages 15–24, 2004.

[3] Haibo Cheng, Peng Zeng, Lingling Xue, Zhao Shi, Peng Wang, and Haibin Yu. Manufacturing Ontology Development based on Industry 4.0 Demonstration Production Line. In *2016 Third International Conference on Trustworthy Systems and their Applications (TSA)*, pages 42–47. IEEE, 2016.

[4] Henrik Bærbak Christensen and Klaus Marius Hansen. An Empirical Investigation of Architectural Prototyping. *Journal of Systems and Software*, 83(1):133–142, 2010.

[5] Infinit website. https://infinit.dk/om-infinit/, 2020.

[6] S. C. Jepsen, T. I. Mørk, J. Hviid, and T. Worm. A Pilot Study of Industry 4.0 Asset Interoperability Challenges in an Industry 4.0 Laboratory. In *2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 571–575, 2020.

[7] Plattform Industrie 4.0. Plattform Industrie 4.0 - What is Industrie 4.0? https://www.plattform-i40.de/PI40/Navigation/EN/Industrie40/WhatIsIndustrie40/what-is-industrie40.htm. Accessed: 2020-11-24.

[8] Emanuel Trunzer, Ambra Calà, Paulo Leitão, Michael Gepp, Jakob Kinghorst, Arndt Lüder, Hubertus Schauerte, Markus Reifferscheid, and Birgit Vogel-Heuser. System architectures for Industrie 4.0 applications. *Production Engineering*, 13(3-4):247–257, 2019.

[9] Jiafu Wan, Shenglong Tang, Di Li, Muhammad Imran, Chunhua Zhang, Chengliang Liu, and Zhibo Pang. Reconfigurable Smart Factory for Drug Packing in Healthcare Industry 4.0. *IEEE Transactions on Industrial Informatics*, 15(1):507–516, 2018.

Author Index