Proceedings of the Federated Conference on Computer Science and Information Systems pp. 579–586 DOI: 10.15439/2017F81 ISSN 2300-5963 ACSIS, Vol. 11

A hybrid method for Optimization Scheduling Groups of Jobs

Jarosław Wikarek, Tadeusz Stefański
Kielce University of Technology Al. 1000-lecia PP 7,
25-314 Kielce, Poland,
Institute of Management and Control Systems
e-mail:{j.wikarek, t.stefanski}@tu.kielce.pl

***Abstract*—This study deals with modelling and optimization of handling jobs (orders) in groups. All jobs in a group should be delivered at the same time after processing. The authors present a novel hybrid method, which includes the modelling and optimization of the problem in the hybrid environment composed of MP (Mathematical Programming) and CLP (Constraint Logic Programming). Due to the large complexity of the optimization problem, dedicated heuristic is also proposed instead of MP. The paper also presents an author's model for optimization scheduling groups of jobs. The model has been implemented in several environments: Hybrid (CLP/MP), Hybrid (CLP, heuristic), MP and heuristic. The obtained results of numerical experiments confirm the high efficiency and usefulness of the hybrid approach to optimize such problems.**

I. Introduction

MANY issues in the area of manufacturing, logistics and services are characterized by handling and processing problems with groups of jobs (orders) and operations, especially when these jobs are to be completed at the same time.

A very good illustration of the handling of jobs (orders) in groups is the process of preparing and serving food in a restaurant [1]. Guests enter the restaurant in different groups at different moments. Each group chooses a table and all jobs of the group members are taken simultaneously. After the accomplishment of these processes, all meal items ordered by a group are served simultaneously. The quality of service and the rate of customer satisfaction are raised if a meal item is served as soon as it is ready. In a restaurant, a group of meal items ordered by guests sitting at a table should be delivered together. Thus, the cooked meal items for a specific group have to wait until the last item of that group is cooked and is ready to be served.

The proposed research problem finds many applications in industrial companies, including but not limited to food, ceramic tile, textile production industries, distributions, supply chain, installation of bulky equipment, manufacturing of complex devices, etc. It can be noticed in many production and logistic industries that have different customers. Assume that each customer has different jobs. Each job has a different handling process function and resources, but all items ordered by a customer or group of customers should be delivered at the same time in one package to reduce the transportation costs, subsequent processing steps time and costs or/and assure proper quality of the product/service and customer satisfaction.

The remainder of the article is organized as follows. Section II presents a literature review. Problem statement, research methodology, mathematical model and contribution are provided in Section III. Computational examples, tests of the implementation platform and discussion are presented in Section IV. Possible extensions of the proposed approach as well as the conclusions are included in Section V

II. Literature Review

To best meet customers' expectations (Section I), multiple decision problems have to be solved. These include processes of food preparation and delivery, proper arrangements of customers at the tables, etc. Due to the number and character of the problems (multimodal, asynchronous, parallel) as well as constraints related to resources, time, etc., they are considered at different decision making levels. At the strategic level, problems of optimal configuration of the order processing/handling environment occur. In the case of a restaurant, these include the selection, configuration and arrangement of tables, known as the Table Mix Problem (TMP) [1,2]. The ''best'' table mix is influenced by several factors such as: the expected number of each size party that will be potential customers; the expected meal duration of each party; the dimensions and the layout of the restaurant, which limit the number and type of tables that can be used, and the possibility of combining tables of different dimensions. Once the TMP is solved, i.e. the number of tables, their size, etc. are decided, it is necessary to assign tables to customers in the most profitable way. Operational decisions are mainly concerned with the most profitable assignment of customers to specific tables. The ''Parties Mix Problem'' consists of deciding on accepting or denying a booking request from different groups of customers, with the aim of maximizing the total expected revenue [3]. The revenue management RM problem is dealt with in multiple papers as the overarching question [3,4]. Scheduling methods for optimal and simultaneous provision of service to groups of customers are proposed most often in the flexible flow-shop system (FFS). In the FFS system, processing is divided into several stages with parallel resources at least in one stage. All of the tasks should pass through all stages in the same order (preparing meals) [5,6]. The exemplified objectives of the problem [6] are minimizing the total amount of time required to complete a group of jobs and minimizing the sum of differences between the completion time of a particular job in the group and the delivery time of this group containing that job (waiting period).

Our motivation was to develop a method that allows problem modeling and optimization for handling incoming jobs in groups with the same date of completion for various forms of organization. Development of optimization models, whose implementation using the proposed method will allow obtaining optimal answers to key questions asked by managers and executive levels.

III. PROBLEM STATEMENT AND METHODOLOGY

The majority of models presented in the literature (Section II) refer to a single problem and optimization according to the set criterion. Fewer studies are devoted to multiple-criteria optimization by operations research (OR) methods [6]. One paper [7] applies constraint programming, but it is used only to solve the static problem of restaurant configuration. Declarative environments such as CLP facilitate problem modeling and introduction of logical and symbolic constraints [8-14]. Unfortunately, high complexity of optimization models and their integer nature contribute to poor efficiency of modeling in OR methods and inefficient optimization in CLP. Therefore, a novel approach to modeling and solving these problems was developed. A declarative environment was chosen as the best structure for this approach [8,10]. Mathematical programming environment was used for problem optimization [15]. This hybrid approach is the basis for the creation of the implementation environment to optimization scheduling groups of jobs. In addition to optimizing particular decision making problems connected with groups of jobs, such environment allows asking various questions while processing the jobs.

The main contribution of this research is the new method for the modeling, support and optimization of decision-making problems for handling jobs in groups. It is based on the integration of CLP and MP/Heuristic. In addition, the linearization and transformation optimization model was built using the CLP environment. Based on the proposed method and model, we designed the framework that allows modeling and optimization the process of handling groups of jobs. The presented method makes it possible to solve the larger size problems in a much shorter time in relation to mathematical programming (MP).

The general concept of hybrid framework (Figure 1) consists in modeling and presolving of a problem in the CLP environment with the final solution (optimization) found in the MP environment or feasible by heuristic algorithm. In all its phases, the platform uses the set of facts having the structure appropriate for the problem being modeled and solved (Figure 2). The set of facts is the informational layer of the framework, which can be implemented as relational database, XML files, etc. Description of the facts for the problem has been shown in Appendix A.

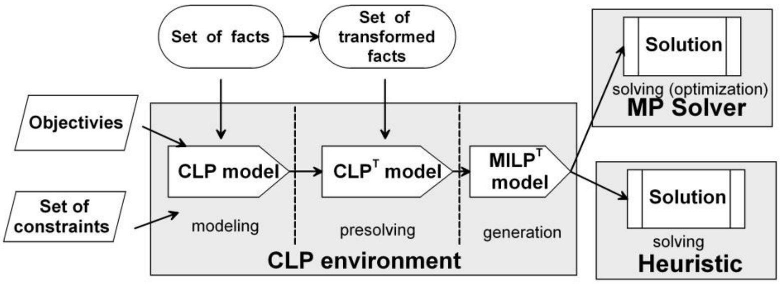

Fig. 1 The concept of hybrid framework

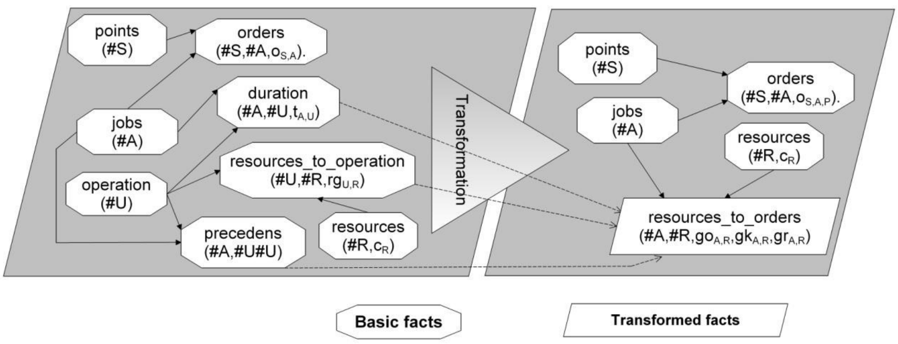

Fig. 2 The scheme of facts for the problem of handling jobs (orders). (#-key attribute of fact)

A. Problem description

This problem can be stated as follows (Figure 3, Table I). Jobs (*a=1..LA*) enter the system in groups. Each jobs consists of operations *(b=1..LB)* and should be processed by specific resources, including parallel resources (*r=1..LR*). The jobs (*a=1..LA*) in each group should be delivered at the same time. It is assumed that all processors in the last stage are eligible to process all jobs. This assumption is valid due to the fact that processors in the last stage (waiters at restaurants who deliver meals or packers in a factory, or quality control) are the same in most of the application areas of the proposed problem. Special points at which orders are submitted and then delivered are introduced /e.g. tables/ (s=*1..LS*). The problem does not cover configuration of the points but relates to handling jobs, as many jobs may come from one customer/jobs several items from the menu/. Each job may be processed by a different resource set in any order.

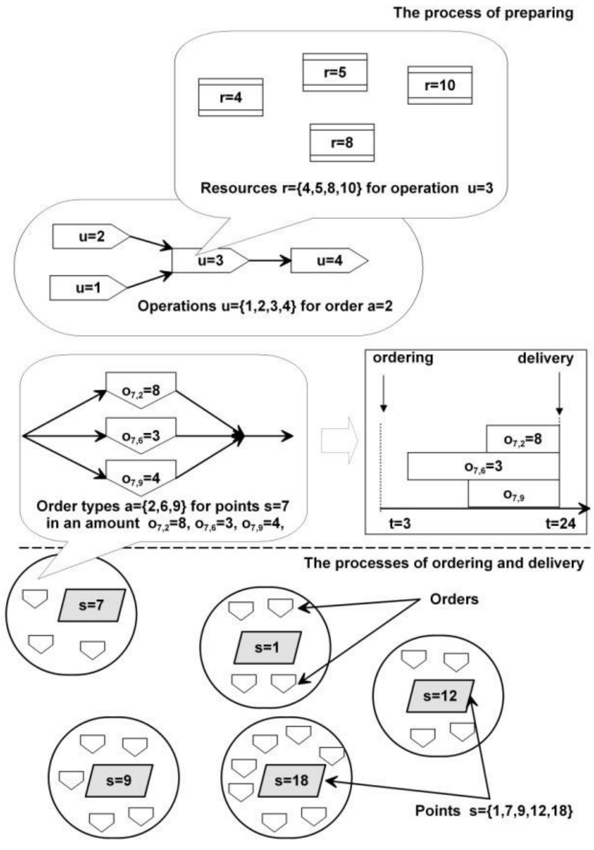

Fig. 3 Scheme for the problem of handling jobs (orders) in a restaurant

The transformation of the problem consisted in the transition from the classical representation in the form of operations to the representation in the form of resources. For this purpose, a corresponding CLP predicates were developed, which based on precedence and resource constraints as well as on the duration of particular operations determined the demand times and sizes for each resource. This transformation allows reductions in the size of the problem by the dimension of operation *u*. The transformation is performed based on the assumption that all the operations for the job are performed without interruption. Transformation is the key element in the hybrid approach. It allows the reduction in the problem size thus reducing combinatorial search space through the reduction of decision variables and constraints (see Appendix B).

TABLE I.
SETS, INDICES, PARAMETERS AND DECISION VARIABLES FOR MATHEMATICAL MODEL

Sets	
Set of points (tables)	LS
Set of jobs (orders)	LA
Set of resources	LR
Set of operations	LU
Number of periods	LT
Indices	
Points (tables)	s=1..LS
Jobs (Orders)	a=1..LA
Resources	r=1..LR
Operation	u=1..LU
Period	t=1..LT
Parameters	
Duration of operation *u* for job (order) *a*	$t_{a,u}$
If the operation u_1 precedes u_2 for job (order) *a* than $kol_{a,u1,u2}$=1 otherwise $kol_{a,u1,u2}$=0	$kol_{a,u1,u2}$
If the operation *u* uses resource *r* than $zas_{u,r}$=1 otherwise $zas_{u,r}$=0	$zas_{u,r}$
Number of *r* resources needed for execution operation *u*	$rg_{u,r}$
The number of available resources *r* in the period *t*.	$cp_{r,t}$
The number of resources of the second type (waiters) available during period *g*	hp_t
The number of jobs (orders) *a* at point *s*	$o_{s,a}$
Decision variables	
Calculated number of periods *t* delivery of all jobs (orders) for point *s*.	F_s
If the execution of operation *u* for job (order) *a* for point *s* uses resource *r* in period *t* then $X_{s,a,r,t}$=1, otherwise $X_{s,a,r,t}$=0	$X_{s,a,u,r,t}$
If *t* is the last period in which resource *r* is used in the execution of operation u for job (order) *a* for point *s* then $Y_{s,a,r,t}=1$, otherwise $Y_{s,a,r,t}$=0	$Y_{s,a,u,r,t}$
Number of period *t* in which operation *u* can be started for job (order) *a* in point *s*	$B_{s,a,u}$
Number of period *t* from resource r can be used for operation *o* of job (order) *a* in point *s*	$S_{s,a,u,r}$
Makespan	C_{max}

Objective functions.

Minimiztion of makespan (1a) or minimization of average waiting time at each point *s* (1b).

Constraints.

Constraint (2) determines the order of execution of operations. Constraint (3) determines the start time of the use of the resource r . Constraint (4) ensures that C_{max} is not less than the time of completion of each operation. Execution time for the point *s* is greater or equal to the time of execution of each jobs *a* at this point (5). Constraint (6) does not allow exceed the available number of resources *r* during period *t*. Constraint (7) provides resource *r* occupancy for the time execution of the operation *u*. Operations cannot be interrupted (8). Simultaneous completion of jobs *a* from the given point is ensured by constraints (9,10). Constraint (11) blocks resources for execution time. Constraint (12) is responsible for the binarity of selected decision variables.

$$\mathrm{Fc1} = \min \mathrm{Cm} \tag{1a}$$

$$\mathrm{Fc2} = \min \frac{1}{LS} \sum_{s=1}^{LL} F_s \tag{2b}$$

$$\begin{gathered} B_{s,a,u_1} + t_{a,u_1} = B_{s,a,u_2} \\ \forall s = 1..LS, a = 1..LA, u_1, u_2 = 1..LU, kol_{a,u_1,u_2} = 1 \end{gathered} \tag{2}$$

$$\begin{gathered} S_{s,a,u,r} = B_{s,a,u} \ \forall s = 1..LS, a = 1..LA, u = 1..LU, \\ r = 1..LR : o_{s,a} > 0, zas_{u,r} > 0 \\ S_{s,a,u,r} = 0 \ \forall s = 1..LS, a = 1..LA, u = 1..L0Ur = 1..LR \\ : o_{s,a} = 0 \\ S_{s,a,u,r} = 0 \ \forall s = 1..LS, a = 1..LA, u = 1..LU, r = 1..LR : \\ zas_{u,r} = 0 \end{gathered} \tag{3}$$

$$C_{max} \ge F_s \ \forall s = 1..LS, : o_{s,a} > 0 \tag{4}$$

$$F_s \ge B_{s,a,u} + t_{a,u} \ \forall a = s..LS, a = 1..LA, u = 1..LU \tag{5}$$

$$\begin{gathered} \sum_{s=1}^{LS} \sum_{a=1}^{LA} \sum_{u=1}^{LU} (X_{s,a,u,r,t} \cdot rg_{u,r} \cdot o_{s,a}) \le cp_{r,t} \\ \forall r = 1..LR, t = 1..LT \end{gathered} \tag{6}$$

$$\begin{gathered} \sum_{t}^{LT} X_{s,a,u,r,t} = t_{a,u} \ \forall s = 1..LS, a = 1..LA, u = 1..LU, \\ r = 1..LR : o_{s,a} > 0, zas_{u,r} > 0 \\ X_{s,a,u,r,t} = 0 \ \forall s = 1..LS, a = 1..LA, u = 1..LU, r = 1..LR : \\ o_{s,a} = 0 \\ X_{s,a,u,r,t} = 0 \ \forall s = 1..LS, a = 1..LA, u = 1..LU, r = 1..LR : \\ zas_{u,r} = 0 \end{gathered} \tag{7}$$

$$\begin{gathered} X_{s,a,u,r,t-1} - X_{s,a,u,r,t} \le Y_{s,a,u,r,t-1} \ \forall s = 1..LS, a = 1..LA, \\ u = 1..LU, r = 1..LR, t = 2..LT : o_{s,a} > 0, zas_{u,r} > 0 \\ Y_{s,a,u,r,t} = 0 \ \forall s = 1..LS, a = 1..LA, u = 1..LU, \\ r = 1..LR, t = LT \\ Y_{s,a,u,r,t} = 0 \ \forall s = 1..LS, a = 1..LA, u = 1..LU, \\ r = 1..LR, t = 1 \end{gathered} \tag{8}$$

$$\begin{gathered} \sum_{t}^{T} Y_{s,a,u,r,t} = 1 \ \forall s = 1..LS, a = 1..LA, u = 1..LU, r = 1..LR \\ : o_{s,a} > 0, zas_{u,r} > 0 \end{gathered} \tag{9}$$

$$\begin{gathered} Y_{s,a,u,r1,t} = Y_{s,a,u,r2,t} \ \forall s = 1..LS, a = 1..LA, u = 1..LU, \\ r1, r2 = 1..R, t = 1..LT, o_{s,a} > 0, zas_{u,r} > 0 \end{gathered} \tag{10}$$

$$X_{s,a,u,r,t} = \begin{cases} 1 & \forall s = 1..LS, a = 1..LA, u = 1..LU, r = 1..LR, \\ & t = 1..LT : t \ge S_{s,a,u,r}, t \le S_{s,a,u,r} + t_{a,u} \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

$$\begin{gathered} X_{s,a,u,r,t} = \{0,1\} \ \forall s = 1..LS, a = 1..LA, u = 1..LU, \\ r = 1..LR, t = 1..LT \\ Y_{s,a,u,r,t} = \{0,1\} \ \forall s = 1..LS, a = 1..LA, u = 1..LU, \\ r = 1..LR, t = 1..LT \end{gathered} \tag{12}$$

B. Transformation

The transformation of the problem consisted in the transition from the classical representation in the form of operations to the representation in the form of resources. For this purpose, a corresponding CLP predicates were developed, which based on precedence and resource constraints as well as on the duration of particular operations determined the demand times and sizes for each resource. This transformation allows reductions in the size of the problem by the dimension of operation *u*. The transformation is performed based on the assumption that all the operations for the job are performed without interruption. Transformation is the key element in the hybrid approach. It allows the reduction in the problem size thus reducing combinatorial search space through the reduction of decision variables and constraints (see Appendix B).

All variables, parameters, auxiliary data etc. (Table II) determined during this process are indicated in the superscript by ^{CLP}.

The mathematical model has been developed, transformed and linearized for the research problem. The sets, indices, parameters, decision variables are presented in Table I.

Objective functions after transformation.

Minimiztion of makespan (1aT) or minimization of average waiting time at each point *s* (1bT).

Constraints after transformation.

Constraint (2T) specifies the moment (period) from which resource *r* is needed to execute job (order) *a*. Constraint (3T) ensures that the makespan is not less than the completion times of all jobs. Constraint (4T) ensures that the number of available resources *r* in period *t* is not exceeded. Constraint (5T) provides resource occupancy for the time of the order execution. Resource *r* is used without interruption during the execution of job (order) *a* from point *s* (6T). Constraint (7T) is for determining decision variable Y. Simultaneous completion of jobs (orders) *a* from the given point is ensured by constraint (8T).

To linearize this model, an ancillary variable was used, $L_{s,t}$={0,1}, determined according to constraint (9T) (where coefficients/factors c^{CLP}_t are determined by the CLP).

TABLE III.
INDICES, PARAMETERS AND DECISION VARIABLES FOR MATHEMATICAL MODEL

Sets	
Set of points (tables)	LS
Set of jobs (orders)	LA
Set of resources	LR
Number of periods	LT
Indices	
Points (tables)	s=1..LS
Jobs (Orders)	a=1..LA
Resources	r=1..LR
Period	t=1..LT
Parameters	
Calculated number of period *t* for the start of demand for resource *r* and job (order) *a (CLP)*	$go^{CLP}{}_{a,r}$
Calculated number of period *t* for the end of demand for resource *r* and job (order) *a (CLP)*	$gk^{CLP}{}_{a,r}$
Number of *r* resources needed for execution of job (order) *a*	$gr^{CLP}{}_{a,r}$
Number used to convert periods to moments (for connecting index *t* with variable $U_{s,a,r}$, if $U_{s,a,r}$=7 then index *t*=7) *(CLP)*	$c^{CLP}{}_{t}$
The number of available resources *r* in the period *t*.	$cp_{r,t}$
The number of resources of the second type (i.e. waiters, packers) available during period *t*	hp_t
Inputs	
The number of jobs (orders) *a* at point *s*	$o_{s,a}$
Decision variables	
Calculated number of periods *t* (using $c^{CLP}{}_{t}$) delivery of all jobs (orders) for point *s*.	F_s
The number of period *t* in which resource *r* can be used for job (order) *a* at point *s*	$U_{s,a,r}$
If the execution of job (order) *a* for point *s* uses resource *r* in period *t* then $X_{s,a,r,t}$=1, otherwise $X_{s,a,r,t}$=0	$X_{s,a,r,t}$
If t is the last period in which resource r is used in the execution of job(order) a for point s then Ys,a,r,t=1, otherwise Ys,a,r,t=0	$Y_{s,a,r,t}$
If *g* is the last period in which jobs (orders) are executed for point *s* then $L_{s,t}$=1, otherwise $L_{s,t}$=0	$L_{s,t}$
makespan	C_{max}

Constraints (10T) and (11T) determine the end of the resource *r* occupancy. Constraint (11T) is an auxiliary constraint responsible for ending the execution of jobs at point *s* but only once. Constraint (12) specifies the number of different type of resources (waiters). Constraint (13) is responsible for the binarity of selected decision variables.

$$Fc1 = \min C_{max} \tag{1aT}$$

$$Fc2 = \min \frac{1}{LS} \sum_{i=1}^{LS} F_s \tag{1bT}$$

$$\begin{gathered} U_{s,a,r} + go^{CLP}_{a,r} = F_s \ \forall\, s = 1..LS, a = 1..LA, r = 1..Lr: \\ o_{s,a} > 0, gr^{CLP}_{a,r} > 0 \\ U_{s,a,r} = 0 \ \forall\, s = 1..LS, a = 1..LA; o_{s,a} = 0 \\ U_{s,a,r} = 0 \ \forall\, s = 1..LS, a = 1..LA: gr^{CLP}_{a,r} = 0 \end{gathered} \tag{2T}$$

$$C_{max} \ge F_s \ \forall\, s = 1..LS; o_{s,a} > 0 \tag{3T}$$

$$\sum_{s=1}^{LS} \sum_{a=1}^{LA} (X_{s,a,r,t} \cdot gr^{CLP}_{a,r} \cdot o_{s,a}) \le cp_{r,t} \ \forall\, r = 1..LR, t = 1..LT \tag{4T}$$

$$\begin{gathered} \sum_{t}^{LT} X_{s,a,r,t} = go^{CLP}_{a,r} - gkk^{CLP}_{a,r} \\ \forall s = 1..LS, a = 1..LA, r = 1..LR : o_{s,a} > 0, gr^{CLP}_{a,r} > 0 \\ X_{s,a,r,t} = 0 \ \forall s = 1..LS, a = 1..LA, r = 1..LR, t = 1..LT \\ : o_{s,a} > 0 \\ X_{s,a,r,t} = 0 \ \forall s = 1..LS, a = 1..LA, r = 1..LR, t = 1..LT \\ : gr^{CLP}_{a,r} > 0 \end{gathered} \tag{5T}$$

$$\begin{gathered} X_{s,a,r,t-1} - X_{s,a,r,t} \le Y_{s,a,r,t-1} \\ \forall\, s = 1..LS, a = 1..LA, r = 1..LR, t = 2..LT \\ : o_{s,a} > 0, gr^{CLP}_{a,r} > 0 \\ Y_{s,a,r,t} = 0 \ s = 1..LS, a = 1..LA, r = 1..LR, t = LG \\ Y_{s,a,r,t} = 0 \ s = 1..LS, a = 1..LA, r = 1..LR, t = 1 \end{gathered} \tag{6T}$$

$$\begin{gathered} \sum_{t}^{LT} Y_{s,a,r,t} = 1 \\ \forall\, s = 1..LS, a = 1..LA, r = 1..LR : o_{s,a} > 0, gr^{CLP}_{a,r} > 0 \end{gathered} \tag{7T}$$

$$\begin{gathered} Y_{s,a,r1,t} = Y_{s,a,r2,t} \ \forall\, s = 1..LS, a = 1..LA, r1, r2 = 1..LR, \\ t = 1..LT : o_{s,a} > 0, gr^{CLP}_{a,r} > 0, gk^{CLP}_{a,r1} = 0, gk^{CLP}_{a,r2} = 0, \end{gathered} \tag{8T}$$

$$F_s = \sum_{t=1}^{LT} c^{CLP}_t L_{s,t} \ \forall\, s = 1..LS \tag{9T}$$

$$\begin{gathered} Y_{s,a,r,t-gk^{CLP}_{a,r}} = L_{s,t} \ \forall\, s = 1..LS, a = 1..LA, r = 1..LR, \\ t = gk^{CLP}_{a,r}..LT : o_{s,a} \ge 0, gr^{CLP}_{a,r} \ge 0 \\ Y_{s,a,r,t} = L_{s,t+gk^{CLP}_{a,r}} \ \forall\, s = 1..LS, A = 1..LA, r = 1..LR, \\ t = 1..LT - gk^{CLP}_{a,r} : o_{s,a} \ge 0, gr^{CLP}_{a,r} \ge 0 \end{gathered} \tag{10T}$$

$$\sum_{t=1}^{LT} L_{s,t} \le 1 \ \forall\, s = 1..LS \tag{11T}$$

$$\sum_{s=1}^{LS} L_{s,t} \le hp_t \ \forall\, t = 1..LT \tag{12T}$$

$$
\begin{gathered}
X_{s,a,r,t} = \{0,1\} \\
\forall\, s = 1..LS, a = 1..LA, r = 1..LR, t = 1..LT \\
Y_{s,a,r,t} = \{0,1\} \\
\forall\, s = 1..LS, a = 1..LA, r = 1..LR, t = 1..LT \\
L_{s,t} = \{0,1\}\ \forall\, s = 1..LS, t = 1..LT
\end{gathered} \tag{13T}
$$

C. Heuristic Algorithm

A heuristic algorithm (Figure 4) was developed to enable solving larger-size problems. Its design was based on the rules of priority and properties. The heuristic algorithm adds consecutive points *s* to the schedule starting with those of the highest priority by the set criteria (Table III). If, while adding *s* point, the algorithm finds that resource constraint is active, leading to the extension of the schedule length, it moves to another step. This step involves checking whether the adjustment of orders using those resources in a given period can provide a better schedule.

Create list L_1 for $s=1..LS$ (for all points for which there are orders)
$C_{max}=0$
L_2 – empty list
Determine the availability of $cp_{r,t}$ resources over time.
While list L_1 is not empty do
    According to a certain criterion *R1* select points *s* from the list L_1
    Schedule jobs according to certain *R2* criterion, assuming maximum resource availability for each point *s*.
    Read $C1_{max}$
    It is not possible to schedule jobs / STOP /.Jobs with such an amount of resources cannot be scheduled. /NO SOLUTION/.
    Add to the list L_2 point *s*
    If $C1_{max} > C_{max}$ then
        $C_{max} = C1_{max}$
    Schedule jobs according to certain *R2* criterion, assuming maximum resource availability for each point s from list L_2 Read $C2_{max}$
    If $C2_{max} > C_{max}$ then
        $C_{max} = C2_{max}$
    Taking into account the actual availability of resources, schedule jobs according to certain *R2* criterion for point *s*. Read $C3_{max}$
    If $C3_{max} > C_{max}$
        Determine the L_3 list of critical resources (their increase decreases the makespan)
        For each resource *r* in the list L_3
            Schedule jobs from the list L_2 starting with the ones that make the most use of the resource r.
            Read $C4_{max}$
            $C3_{max} > C4_{max}$ then
                $C3_{max} = C4_{max}$
                Correct the availability of resources to suit selected schedule.
    Reduce the availability of resources in each period by the amount needed to execute the orders from point *s*
    Remove from list L1 point s
STOP – feasible solution found

Fig. 4 The heuristic algorithm dedicated to the scheduling group of jobs

TABLE III.
POSSIBLE VALUES OF CRITERION

Criterion	Description
R1	• The order that uses the most critical resource • The order with the longest execution time
R2	Queue priority methods known from literature • LPT • SPT

Underlining indicated the criteria chosen for the computational experiments (Section IV).

IV. NUMERICAL EXPERIMENTS

In order to verify and evaluate the proposed approach and models, many numerical experiments were performed for optimization scheduling groups of jobs. All the experiments relate to the problem with fifty points (*s=1..50*), twenty order types (*a=1..20*), fifty resource types (*r=1..50*), fifteen operation types (u=1..15) and from three to one hundred fifty orders $o_{S,A}$

The main part of the study was a comparative analysis performed for Fc1 and Fc2 in four environments: mathematical programming (MP), heuristic algorithm, hybrid1 (CLP&MP) and hybrid2 (CLP& heuristic algorithm) hybrid and MP) to evaluate the effectiveness and efficiency of the proposed hybrid approach relative to the classical MP environment and heuristic algorithm. The experiments for examples E1..E7 were conducted for various values of parameters *LS,N*. The results are included in Appendix B (Table B1, Table B2). The application of the hybrid approach leads to a substantial reduction in (i) number of decision variables (up to fifteen times), (ii) number of constraints (up to two times) (iii) computing time (more than twenty times faster) for the above examples. For larger numerical examples, such as E3..E7 the MP-based approach cannot be used due to the length of calculations and, most importantly, exceeded size of the problems accepted by the available MP solvers. Using hybrid approach (hybrid2), it reduces the computation time twice and improves the quality of approximate solutions (0-1% worse from optimal) in relation to the use heuristic algorithm (the quality of approximate solutions are 1-2% worse from optimal).

V. CONCLUSION

The proposed approach to the modelling and optimization scheduling groups of jobs can be used in many areas. Similar issues exist wherever there are a variety of customer jobs (orders), the handling of which require processes and additionally, both are ordered and executed jointly with a single delivery deadline. In practice, such an approach to group job (order) handling occurs in manufacturing, services, logistics and project management. The presented framework, which is an implementation of the proposed approach, enables effective optimization scheduling groups of jobs. This allows the implementation of optimization models with different objective functions and the introduction of additional constraints to the models already implemented. The illustrative example shows only part of the framework's potential. Significant results are to increase both the speed and the size of the problems solved.

It is foreseen in further research the use of a hybrid approach to (a) modeling and solving scheduling problems in production [16,17], (b) modeling and optimization of IoT processes [18], and (c) implementation of more complex models, uncertainty, fuzzy logic etc..

APPENDIX A

TABLE A1.
DESCRIPTION OF FACTS

Name	Description
points(#S)	A fact that describes the points.
jobs(#A)	A fact that describes the type of jobs (orders).
operations(#U)	A fact that describes the type of operations.
precedens(#A,#U,#U)	A fact that describes the precedence operations in job (order)..
duration(#A,#U,$t_{A,U}$)	A fact that describes execution time for operations in job.
resources(#R,c_R)	A fact that describes resources (the number of each type)
resource_to_operation(#U,#R,$rg_{U,R}$)	A fact that specifies acceptable allocation of resources to operations.
orders(#S,#A,$o_{S,A}$)	A fact that describes orders at point
resources_to_orders(#A, #R,$go_{A,R}$, $gk_{A,R}$, $go_{A,R}$)	A fact determines what resources are needed to complete the order .

APPENDIX B

TABLE B1.
THE RESULTS OF NUMERICAL EXPERIMENTS FOR EXAMPLES WITH FC1

E	NS	N	Primary model						Transformed model					
			MP				Heuristic		Hybrid1				Hybrid2	
			V_{int}	C	Fc1	T	Fc1	T	V_{int}	C	Fc1	T	Fc1	T
E1	1	3	365	288	28	10	28	4	24	183	28	5	28	3
E2	5	17	10330	6792	45	234	45	23	689	5130	45	89	45	16
E3	10	36	43751	28044	97	546	97	33	2917	21690	97	124	97	21
E4	20	54	131252	83052	185*	900**	182	39	8750	65016	179	234	180	28
E5	30	100	364590	229700	254*	900**	244	48	24306	180550	240	548	242	31
E6	40	130	631956	397280	NFSF	900**	310	56	42130	312910	310	754	310	34
E7	50	150	911475	572250	NFSF	900**	345	64	60765	451275	342	834	344	36

E	Experiments
NS	Number of points
N	Total number of jobs
T	Time of finding solution (in seconds)
V_{int}	The number of decision variables
C	The number of constrains
*	Feasible solution (not found optimality)
**	Interrupt the process of finding a solution after a given time 900 s

TABLE B2.
THE RESULTS OF NUMERICAL EXPERIMENTS FOR EXAMPLES WITH FC2

E	NS	N	Primary model						Transformed model					
			MP				Heuristic		Hybrid1				Hybrid2	
			V_{int}	C	Fc2	T	Fc2	T	V_{int}	C	Fc2	T	Fc2	T
E1	1	3	365	288	28	10	28	4	24	183	28	5	28	3
E2	5	17	10330	6792	32,4	232	32,4	21	689	5130	32,4	81	32,4	14
E3	10	36	43751	28044	52,1	546	52,1	33	2917	21690	52,1	124	52,1	21
E4	20	54	131252	83052	98,95	594	101,23	32	8750	65016	98,95	212	98,95	24
E5	30	100	364590	229700	154,6*	900**	142,4	42	24306	180550	138,4	522	142,4	29
E6	40	130	631956	397280	NFSF	900**	198,2	62	42130	312910	192,4	647	192,4	32
E7	50	150	911475	572250	NFSF	900**	212,2	72	60765	451275	209,4	734	209,4	36

E	Experiments
NS	Number of points
N	Total number of jobs
T	Time of finding solution (in seconds)
V_{int}	The number of decision variables
C	The number of constrains
*	Feasible solution (not found optimality)
**	Interrupt the process of finding a solution after a given time 900 s

REFERENCES

[1] F. Guerriero, G. Miglionico, F. Olivito, "Strategic and operational decisions in restaurant revenue management", *European Journal of Operational Research 237*,2014, pp. 1119–1132.

[2] G.M. Thompson, "Optimizing restaurant table configuration: Specifying combinable tables", *Cornell Hotel and Restaurant Administration Quartely 44*, 2003, pp. 53–60.

[3] W. C. Chiang, J.C.H Chen, X. Xu, "An overview of research on revenue management: Current issues and future research", *International Journal of Revenue Management 1,*2007 pp.97–128.

[4] G. M. Thompson, "Restaurant profitability management, The evolution of restaurant revenue management", *Cornell Hospitality Quarterly,* 51, 2010, pp. 308–322.

[5] I. Ribas, R. Leisten, JM. Framinan, "Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective", *Computer Operation Research ,* 37, 2010, pp. 1439–1454.

[6] B. Tadayon, N. Salmasi, "A two-criteria objective function flexible flowshop scheduling problem with machine eligibility constraint" *The International Journal of Advanced Manufacturing Technology,* 64(5-8), 2013, pp. 1001-1015.

[7] A. Vidotto, K.N. Brown, J.C Beck, "Managing Restaurant Tables using Constraints" *Knowledge Based Systems,* March, 20(2), 2007, pp. 160-169.

[8] K. Apt, M. Wallace, "Constraint Logic Programming using Eclipse". *Cambridge: Cambridge University Press, 2006*.

[9] P. Sitek, J. Wikarek, "A hybrid framework for the modelling and optimisation of decision problems in sustainable supply chain management", *International Journal of Production Research,* vol 53(21), 2015, pp 6611-6628, doi:10.1080/00207543. 2015.1005762

[10] P. Sitek, J. Wikarek, "A Hybrid Programming Framework for Modeling and Solving Constraint Satisfaction and Optimization Problems" *Scientific Programming*, vol. 2016, Article ID 5102616, 2016. doi:10.1155/2016/5102616.

[11] P. Sitek, "A hybrid approach to the two-echelon capacitated vehicle routing problem (2E-CVRP)", *Advances in Intelligent Systems and Computing,* 267, 2014, pp. 251–263, DOI: 10.1007/978-3-319-05353-0_25

[12] J. Wikarek, :Implementation Aspects of Hybrid Solution Framework", *Recent Advancees in Automation, Robotics and Measuring Techniques* vol 267, 2014, pp. 317-328. doi: 10.1007/978-3-319-05353-0_31

[13] G. Bocewicz, I. Nielsen, Z. Banaszak, "Iterative multimodal processes scheduling" *Annual Reviews in Control 38(1)*, 2014, pp. 113-132

[14] M. Relich, "Knowledge acquisition for new product development with the use of an ERP database", *Proceedings of the Federated Conference on Computer Science and Information Systems,* 2013, pp. 1285–1290.

[15] A. Schrijver, A. "Theory of Linear and Integer Programming", John Wiley & Sons, New York, NY, USA 1998.

[16] Z. Li, M.N Janardhanan, Q. Tang, P. Nielsen, "Co-evolutionary particle swarm optimization algorithm for two-sided robotic assembly line balancing problem", *Advances in Mechanical Engineering*, 8 (9), 2016, pp. 1-14. doi: http://dx.doi.org/10.1177/1687814016667907

[17] I. Nielsen, Q. Dang, P. Nielsen, P Pawlewski, "Scheduling of mobile robots with preemptive tasks", *Advances in Intelligent Systems and Computing*, 290, 2014 pp. 19-27. doi:https://doi.org/10.1007/978-3-319-07593-8_3

[18] S. Deniziak, T. Michno, P. Pieta, "IoT-Based Smart Monitoring System Using Automatic Shape Identification", *Advances in Intelligent Systems and Computing book series* (AISC, volume 511), 2015, pp. 1-18, doi:https://doi.org/10.1007/978-3-319-46535-7_1