

A hybrid method for modeling and solving constrained search problems

Paweł Sitek

Kielce University of Technology
Al. 1000-lecia PP 7, 25-314
Kielce, Poland, Institute of Man-
agement and Control Systems
e-mail:sitek@tu.kielce.pl

Jarosław Wikarek

Kielce University of Technology
Al. 1000-lecia PP 7, 25-314
Kielce, Poland, Institute of Man-
agement and Control Systems
e-mail:j.wikarek@tu.kielce.pl

Abstract—The paper presents a concept and the outline of the implementation of a hybrid approach to modeling and solving constrained problems. Two environments of mathematical programming (MP) and logic programming (LP) were integrated. The strengths of integer programming (IP) and constraint logic programming (CLP), in which constraints are treated in a different way and different methods are implemented, were combined to use the strengths of both. The proposed approach is particularly important for the decision models with an objective function and many discrete decision variables added up in multiple constraints.

To validate the proposed approach, two illustrative examples are presented and solved. The first example is the authors' original model of cost optimization in the supply chain with multimodal transportation. The second one is the two-echelon variant of the well-known Capacitated Vehicle Routing Problem, 2E-CVRP.

I. INTRODUCTION

THE vast majority of models [1]–[4] of decision support and/or optimization in manufacturing, distribution, supply chain management, etc., have been formulated as the mixed integer linear programming (MILP) or integer programming (IP) problems and solved using the operations research (OR) methods. Their structures are similar and proceed from the principles and requirements of mathematical programming. The constraint-based environments have the advantage over traditional methods of mathematical modeling in that they work with a much broader variety of interrelated constraints (resource, time, technological, and financial) and allow producing “natural” solutions for highly combinatorial problems.

A. Constraint-based environments

We strongly believe that the constraint-based environment [5]–[7] offers a very good framework for representing the knowledge and information needed for the decision support. The central issue for a constraint-based environment is a constraint satisfaction problem. Constraint satisfaction problems (CSPs) are the mathematical problems defined as a set of elements whose state must satisfy a number of constraints. CSPs represent the entities in a problem as a homogeneous collection of finite constraints over variables, which are solved using constraint satisfaction methods. CSPs are

the subject of intense study in both artificial intelligence and operations research, since the regularity in their formulation provides a common basis for analyzing and solving the problems of many unrelated families [5]. Formally, a constraint satisfaction problem is defined as a triple (X, D, C) , where X is a set of variables, D is a domain of values, and C is a set of constraints. Every constraint is in turn a pair (t, R) (usually represented as a matrix), where t is an n -tuple of variables and R is an n -ary relation on D . An evaluation of the variables is a function from the set of variables to the domain of values, $v: X \rightarrow D$. An evaluation v satisfies constraint $((x_1, \dots, x_n), R)$ if $(v(x_1), \dots, v(x_n)) \in R$. A solution is an evaluation that satisfies all constraints.

Constraint satisfaction problems on finite domains are typically solved using a form of search. The most widely used techniques include variants of backtracking, constraint propagation, and local search. Constraint propagation embeds any reasoning that consists in explicitly forbidding values or combinations of values for some variables of a problem because a given subset of its constraints cannot be satisfied otherwise [26].

CSPs are frequently used in constraint programming. Constraint programming is the use of constraints as a programming language to encode and solve problems.

Constraint logic programming (CLP) is a form of constraint programming (CP), in which logic programming is extended to include concepts from constraint satisfaction. A constraint logic program is a logic program that contains constraints in the body of clauses. Constraints can also be present in the goal. These environments are declarative.

The declarative approach and the use of logic programming provide incomparably greater possibilities for decision problems modeling than the pervasive approach based on mathematical programming.

B. Paper contents

In this paper we focus on the problem of modeling and solving decision problems using the novel hybrid approach. Having combined the strengths of MILP and CP/CLP (II, III), we developed the environment that ensures the better and easier way of problem modeling and implementation and that provides the more effective search solution (IV). In

order to verify the proposed approach, two illustrative examples are presented (V).

II. MOTIVATION

Based on [1]–[4], and our previous work [6], [8]–[12], we observed some advantages and disadvantages of these environments.

An integrated approach of constraint programming (CP) and mixed integer programming (MIP) can help to solve optimization problems that are intractable with either of the two methods alone [13]–[16]. Although operations research (OR) and constraint programming (CP) have different roots, the links between the two environments have grown stronger in recent years.

Both MIP/MILP/IP and finite domain CP/CLP involve variables and constraints. However, the types of the variables and constraints that are used, and the way the constraints are solved, are different in the two approaches [16].

MILP relies completely on linear equations and inequalities in integer variables, i.e., there are only two types of constraints: linear arithmetic (linear equations or inequalities) and integrity (stating that the variables have to take their values in the integer numbers). In finite domain CP/CLP, the constraint language is richer. In addition to linear equations and inequalities, there are various other constraints: disequalities, nonlinear, symbolic (*alldifferent*, *disjunctive*, *cumulative* etc).

The motivation behind this work was to create a hybrid approach for supply chain modeling and optimization instead of using integer programming or constraint programming separately. We developed the hybrid framework for modeling and optimization of supply chain problems. In both MILP/MIP and CP/CLP, there is a group of constraints that can be solved with ease and a group of constraints that are difficult to solve. The easily solved constraints in MILP/MIP are linear equations and inequalities over rational numbers.

Integrity constraints are difficult to solve using mathematical programming methods and often the real problems of MIP / MILP make them NP-hard.

In CP/CLP, domain constraints with integers and equations between two variables are easy to solve. The system of such constraints can be solved over integer variables in polynomial time. The inequalities between two variables, general linear constraints (more than two variables), and symbolic constraints are difficult to solve, which makes real problems in CP/CLP NP-hard. This type of constraints reduces the strength of constraint propagation. As a result, CP/CLP is incapable of finding even the first feasible solution.

It follows from the above that what is difficult to solve in one environment can be easy to solve in the other.

The motivation was to offer the most effective tools for model-specific constraints and solution efficiency.

III. STATE OF THE ART

As mentioned in Chapter I, the vast majority of decision-making models for the problems of production, logistics, supply chain are formulated in the form of mathematical programming (MIP, MILP, IP).

Due to the structure of these models (summing of discrete decision variables in the constraints and the objective function) and a large number of discrete decision variables (integer and binary) they can only be applied to small problems. Another disadvantage is that only linear constraints can be used. In practice, the issues related to the production, distribution and supply chain constraints are often logical, nonlinear, etc. For these reasons the problem was formulated in a new way,

In our approach to modeling and optimization of constrained search problems we proposed the optimization environment, where:

- knowledge related to the problem can be expressed as linear and logical constraints (implementing all types of constraints of the previous MILP/MIP models [10]–[14] and introducing new types of constraints (logical, nonlinear, symbolic etc.));
- the optimization model solved using the proposed framework can be formulated as a pure model of MILP/MIP or of CP/CLP, or it can also be a hybrid model;
- the problem is modeled in CP/CLP, which is far more flexible than MIP/MILP/IP;
- the novel method of constraint propagation is introduced (obtained by transforming the optimization model to explore its structure);
- constrained domains of decision variables, new constraints and values for some variables are transferred from CP/CLP into MILP/MIP;
- the efficiency of finding solutions to the problems of larger sizes is increased.

As a result, we obtained the more effective search solution for a certain class of decision and optimization constrained problems.

IV. HYBRID SOLUTION ENVIRONMENT

Both environments have advantages and disadvantages. Environments based on the constraints such as CLPs are declarative and ensure a very simple modeling of decision problems, even those with poor structures if any. The problem is described by a set of logical predicates. The constraints can be of different types (linear, non-linear, logical, binary, etc.). The CLP does not require any search algorithms. This feature is characteristic of all declarative backgrounds, in which modeling of the problem is also a solution, just as it is in Prolog, SQL, etc. The CLP seems perfect for modeling and solving any decision problem.

In OR numerous models of decision-making have been developed and tested, particularly in the area of decision optimization. Constantly improved methods and mathematical programming algorithms, such as the simplex algorithm,

branch and bound, branch-and-cost [20] etc., have become classics now.

The proposed method's strength lies in high efficiency of optimization algorithms and a substantial number of tested models. The decision problems we deal with in this paper, very common in manufacturing, logistics, supply chain, etc., have a number of decision variables, including binary and integer ones, which are aggregated in the constraints.

Traditional methods when used alone to solve complex problems provide unsatisfactory results. This is related directly to different treatment of variables and constraints in those approaches (II). The proposed hybrid approach, a composition of methods as described in Chapter III offers the optimal system for specific contexts.

A. Architecture and Implementation of Hybrid Solution Environment

This Hybrid Solution Environment (HSE) consists of MIP/MILP/CLP/Hybrid models and Hybrid Solution Framework (FSF) to solve them (Fig. 1). The concept of this framework with its phases (P1 .. P5, G1 .. G3) is presented in Fig. 2.

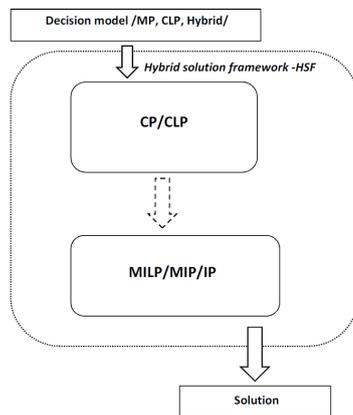


Fig. 1 Scheme of the hybrid solution environment (HSE)

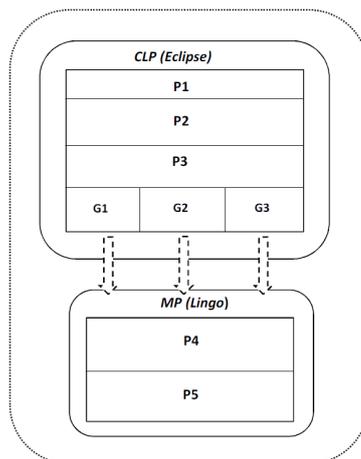


Fig. 2 Detailed scheme of the hybrid solution framework (HSF)

A detailed description of the phases in the order of execution is shown in Table I.

From a variety of tools for the implementation of the CP/CLP in HSE, ECLIPSE software [21] was selected. ECLIPSE is an open-source software system for the cost-effective development and deployment of constraint programming applications. Environment for the implementation of MILP/MIP/IP in HSE was LINGO by LINDO Systems. LINGO Optimization Modeling Software is a powerful tool for building and solving mathematical optimization models [22].

TABLE I
DESCRIPTION OF PHASES

Phase	P1
Name	Implementation of decision model
Description	The implementation of the model in CLP, the term representation of the problem in the form of predicates.
Phase	P2
Name	Transformation of implemented model for better constraint propagation (optional)
Description	The transformation of the original problem aimed at extending the scope of constraint propagation. The transformation uses the structure of the problem. The most common effect is a change in the representation of the problem by reducing the number of decision variables, and the introduction of additional constraints and variables, changing the nature of the variables, etc.
Phase	P3
Name	Constraint propagation
Description	Constraint propagation for the model. Constraint propagation is one of the basic methods of CLP. As a result, the variable domains are narrowed, and in some cases, the values of variables are set, or even the solution can be found.
Phase	G1
Name	Generation of MILP/MIP/IP model
Description	Generation of the model for mathematical programming. Generation performed automatically using CLP predicate. The resulting model is in a format accepted by the system LINGO.
Phase	G2
Name	Generation of additional constraints (optional)
Description	Generation of additional constraints on the basis of the results obtained in step P3
Phase	G3
Name	Generation domains of decision variables and other values
Description	Generation of domains for different decision variables and other parameters based on the propagation of constraints. Transmission of this information in the form of fixed value of certain variables and/or additional constraints to the MP.
Phase	P4
Name	Merging MILP/MIP/IP model
Description	Merging files generated during the phases G1, G2, G3 into one file. It is a model file format in LINGO system.
Phase	P5
Name	Solving MILP/MIP/IP model
Description	The solution model from the previous stage by LINGO. Generation of the report with the results and parameters of the solution.

ECLiPS^e software is the environmental leader in HSE. ECLiPS^e was used to implement the following phases of the framework: P1, P2, P3, G1, G2, G3 (Fig. 2, Table I). The transformed files of the model were transferred from ECLiPS^e to LINGO where they were merged (P4). Then the complete model was solved using LINGO efficient solvers (P5). Constraint propagation (phase–P3) greatly affected the efficiency of the solution. Therefore phase P2 was introduced. During this phase, the transformation was performed using the structure and properties of the model. This is an optional phase that depends on the modeled problem. The details of this phase will be presented in one of the illustrative examples in Chapter V (cost optimization of supply chain).

V. ILLUSTRATIVE EXAMPLES

The proposed HSE environment was verified and tested for two illustrative examples. The first example is the authors' original model of cost optimization of supply chain with multimodal transport (section A). The second is a 2E-CVRP model (section B). It is the known benchmark of a very large number of sets/instances of data and their solutions.

A. Cost optimization of supply chain with multimodal transport

A detailed description of the cost optimization of supply chain models, their constraints, parameters and decision variables etc. are presented in [17] and Table II.

During the first stage, the model was formulated as a MILP problem [9], [10], [17] in order to test the proposed environment (Fig. 1,2) against the classical integer-programming environment [22]. The next step involved the implementation and solving of the hybrid model. Indices, parameters and decision variables in the models together with their descriptions are provided in Table II. The simplified structure of the supply chain network for this model, composed of producers, distributors and customers is presented in Fig.3.

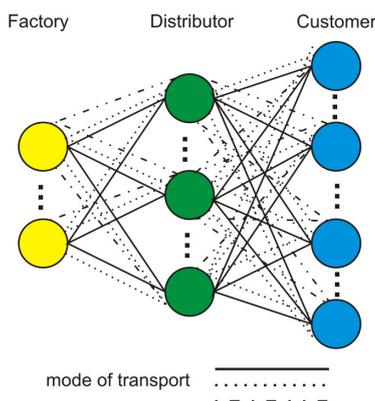


Fig. 3 The simplified structure of the supply chain network

The proposed models are the cost models that take into account three other types of parameters, i.e., the spatial parameters (area/volume occupied by the product, distributor capacity and capacity of transport unit), time (duration of de-

TABLE II
SUMMARY INDICES, PARAMETERS AND DECISION VARIABLES

Symbol	Description
Indices	
k	product type (k=1..O)
j	delivery point/customer/city (j=1..M)
i	manufacturer/factory (i=1..N)
s	distributor /distribution center (s=1..E)
d	mode of transport (d=1..L)
N	number of manufacturers/factories
M	number of delivery points/customers
E	number of distributors
O	number of product types
L	number of mode of transport
Input parameters	
F_s	the fixed cost of distributor/distribution center s
P_k	the area/volume occupied by product k
V_s	distributor s maximum capacity/volume
$W_{i,k}$	production capacity at factory i for product k
$C_{i,k}$	the cost of product k at factory i
$R_{s,k}$	if distributor s can deliver product k then $R_{sk}=1$, otherwise $R_{sk}=0$
$T_{p_{s,k}}$	the time needed for distributor s to prepare the shipment of product k
$T_{c_{j,k}}$	the cut-off time of delivery to the delivery point/customer j of product k
$Z_{j,k}$	customer demand/order j for product k
Z_{t_d}	the number of transport units using mode of transport d
P_{t_d}	the capacity of transport unit using mode of transport d
$T_{f_{i,s,d}}$	the time of delivery from manufacturer i to distributor s using mode of transport d
$K1_{i,s,k,d}$	the variable cost of delivery of product k from manufacturer i to distributor s using mode of transport d
$R1_{i,s,d}$	if manufacturer i can deliver to distributor s using mode of transport d then $R1_{i,s,d}=1$, otherwise $R1_{i,s,d}=0$
$A_{i,s,d}$	the fixed cost of delivery from manufacturer i to distributor s using mode of transport d
$Koa_{i,s,d}$	the total cost of delivery from manufacturer i to distributor s using mode of transport d
$T_{m_{s,j,d}}$	the time of delivery from distributor s to customer j using mode of transport d
$K2_{s,j,k,d}$	the variable cost of delivery of product k from distributor s to customer j using mode of transport d
$R2_{s,j,d}$	if distributor s can deliver to customer j using mode of transport d then $R2_{s,j,d}=1$, otherwise $R2_{s,j,d}=0$
$G_{s,j,d}$	the fixed cost of delivery from distributor s to customer j using mode of transport d
$Kog_{s,j,d}$	the total cost of delivery from distributor s to customer j using mode of transport d
Od_d	the environmental cost of using mode of transport d
Decision variables	
$X_{i,s,k,d}$	delivery quantity of product k from manufacturer i to distributor s using mode of transport d
$Xa_{i,s,d}$	if delivery is from manufacturer i to distributor s using mode of transport d then $Xa_{i,s,d}=1$, otherwise $Xa_{i,s,d}=0$
$Xb_{i,s,d}$	the number of courses from manufacturer i to distributor s using mode of transport d
$Y_{s,j,k,d}$	delivery quantity of product k from distributor s to customer j using mode of transport d
$Ya_{s,j,d}$	if delivery is from distributor s to customer j using mode of transport d then $Ya_{s,j,d}=1$, otherwise $Ya_{s,j,d}=0$
$Yb_{s,j,d}$	the number of courses from distributor s to customer j using mode of transport d
Tc_s	if distributor s participates in deliveries, then $Tc_s=1$, otherwise $Tc_s=0$
CW	Arbitrarily large constant

livery and service by distributor, etc.) and the transport mode.

The main assumptions made for the construction of these models were as follows:

- the shared information process in the supply chain consists of resources (capacity, versatility, costs), inventory (capacity, versatility, costs, time), production (capacity, versatility, costs), product (volume), transport (cost, mode, time), demand, etc;
- a part of the supply chain has the structure as in Fig. 3.;
- the transport is multimodal (several modes of transport, a limited number of means of transport for each mode);
- the environmental aspects of use of transport modes are taken into account;
- different products are combined in one batch of transport;
- the cost of supplies is presented in the form of a function (in this approach, linear function of fixed and variable costs);
- models have linear or linear and logical (hybrid model) constraints;
- logical constraints of hybrid model allow the distribution of exclusively one of two selected products in the distribution center and allow the production of exclusively one of two selected products in the factory.

Details of both mathematical models for cost optimization of supply chain are presented in [17].

Objective function

The objective function defines the aggregate costs of the entire chain and consists of five elements. The first element comprises the fixed costs associated with the operation of the distributor involved in the delivery (e.g. distribution centre, warehouse, etc.). The second element corresponds to environmental costs of using various means of transport. Those costs are dependent on the number of courses of the given means of transport, and on the other hand, on the environmental levy, which in turn may depend on the use of fossil fuels and carbon-dioxide emissions.

The third component determines the cost of the delivery from the manufacturer to the distributor. Another component is responsible for the costs of the delivery from the distributor to the end user (the store, the individual client, etc.). The last component of the objective function determines the cost of manufacturing the product by the given manufacturer.

Formulating the objective function in this manner allows comprehensive cost optimization of various aspects of supply chain management. Each subset of the objective function with the same constraints provides a subset of the optimization area and makes it much easier to search for a solution.

Constraints

The model was based on constraints (2) .. (24) Constraint (2) specifies that all deliveries of product k produced by the manufacturer i and delivered to all distributors s using mode

of transport d do not exceed the manufacturer's production capacity.

Constraint (3) covers all customer j demands for product k ($Z_{j,k}$) through the implementation of delivery by distributors s (the values of decision variables $Y_{i,s,k,d}$). The flow balance of each distributor s corresponds to constraint (4). The possibility of delivery is dependent on the distributor's technical capabilities - constraint (5). Time constraint (6) ensures the terms of delivery are met. Constraints (7a), (7b), (8) guarantee deliveries with available transport taken into account.

The hybrid model was additionally enriched with logical constraints.

First logical constraint allows the distribution of exclusively one of the two selected products in the distribution center s . Second logical constraint allows the production of exclusively one of the two selected products in the factory i .

These constraints stem from technological, marketing, sales or safety restrictions. Therefore, some products cannot be distributed and/or produced together. The constraint can be re-used for different pairs of product k and for some or all of the distribution centers s and factories i . A logical constraint like this cannot be easily implemented in a MILP model.

Model transformation

Due to the nature of the decision problem (adding up decision variables and constraints involving a lot of variables), the constraint propagation efficiency decreases dramatically. Constraint propagation is one of the most important methods in CLP affecting the efficiency and effectiveness of the CLP and hybrid optimization environment (Fig. 1, Table I). For that reason, research into more efficient and more effective methods of constraint propagation was conducted. The results included different representation of the problem and the manner of its implementation.

The classical problem modeling in the CLP environment consists in building a set of predicates with parameters.

Each CLP predicate has a corresponding multi-dimensional vector representation. While modeling both problems, quantities i, s, k, d and decision variable $X_{i,s,k,d}$ were vector parameters (Fig. 4a). As shown in Fig. 4b, for each vector there were 5 values to be determined, defining the size of the delivery, factories, distributors involved in the delivery and the mode of transport.

[Z_n,P,M,D,F,Tu,Tu,Oq,X,T]

Fig. 4a Representation of the problem in the classical approach-definition

**[[z_1,p1,m1,_,_,_,10,_,8],
[z_2,p1,m2,_,_,_,20,_,6],...]**

Fig. 4b Representation of the problem in the classical approach-process of finding a solution

The process of finding the solution may consist in using the constraint propagation methods, variable labeling and the

backtracking mechanism. The numbers of parameters that must be specified/labeled in the given predicate/vector critically affect the quality of constraint propagation and the number of backtracks. In both models presented above, the classical problem representation included five parameters: i , s , k , d and $X_{i,s,k,d}$. Considering the domain size of each parameter, the process was complex and time-consuming. In addition, the above representation (Fig. 4a, Fig. 4b) arising from the structure of the problem is the cause of many backtracks.

Our idea involved the transformation of the problem by changing its representation without changing the very problem. All permissible routes were first generated based on the fixed data and a set of orders, then the specific values of parameters i , s , k , d were assigned to each of the routes. In this way, only decision variables $X_{i,s,k,d}$ (deliveries) had to be specified (Fig. 5). This transformation fundamentally improved the efficiency of the constraint propagation and reduced the number of backtracks. A route model is a name adopted for the models that underwent the transformation.

[[name_1,f1,p1,c1,m1,s1,s1,5,12,100,_],
[name_2,f1,p1,c1,m1,s1,s2,6,14,100,_],
[name_3,f1,p1,c1,m1,s2,s1,6,22,100,_],...

Fig. 5 Representation of the problem in the novel approach- set of feasible routes

Symbols necessary to understand both the representation of the problem and their descriptions are presented in Table III.

TABLE III
SYMBOLS USED IN THE REPRESENTATION OF THE PROBLEM

Symbol	Description
Z_n	order number
P	products, $P \in \{p_1, p_2, \dots, p_o\}$
M	customers, $M \in \{m_1, m_2, \dots, m_m\}$
D	distributors, $D \in \{c_1, c_2, \dots, c_e\}$
F	factories, $F \in \{f_1, f_2, \dots, f_n\}$
Tu	transport unit, $Tu \in \{s_1, s_2, \dots, s_l\}$
T	delivery time/period
Oq	order quantity
X	delivery quantity
Name_	routes name-number

B. Two-Echelon Capacitated Vehicle Routing Problem

The 2E-CVRP is proposed as a benchmark verifying the presented approach. The Two-Echelon Capacitated Vehicle Routing Problem (2E-CVRP) is an extension of the classical Capacitated Vehicle Routing Problem (CVRP) where the delivery depot-customers pass through intermediate depots (called satellites). As in CVRP, the goal is to deliver goods to customers with known demands, minimizing the total delivery cost in the respect of vehicle capacity constraints. Multi-echelon systems presented in the literature usually explicitly consider the routing problem at the last level of the transportation system, while a simplified routing problem is considered at higher levels [18], [19], [23].

In 2E-CVRP, the freight delivery from the depot to the customers is managed by shipping the freight through intermediate depots. Thus, the transportation network is decomposed into two levels (Fig. 6): the 1st level connecting the depot (d) to intermediate depots (s) and the 2nd one connecting the intermediate depots (s) to the customers (c). The objective is to minimize the total transportation cost of the vehicles involved in both levels. Constraints on the maximum capacity of the vehicles and the intermediate depots are considered, while the timing of the deliveries is ignored.

From a practical point of view, a 2E-CVRP system operates as follows (Fig. 6):

- freight arrives at an external zone, the depot, where it is consolidated into the 1st-level vehicles, unless it is already carried into a fully-loaded 1st-level vehicles;
- each 1st-level vehicle travels to a subset of satellites that will be determined by the model and then it will return to the depot;
- at a satellite, freight is transferred from 1st-level vehicles to 2nd-level vehicles;

The mathematical model (MILP) was taken from [17]. It required some adjustments and error corrections. Table IV shows the parameters and decision variables of 2E-CVRP. Figure 6 shows an example of the 2E-CVRP - transportation network.

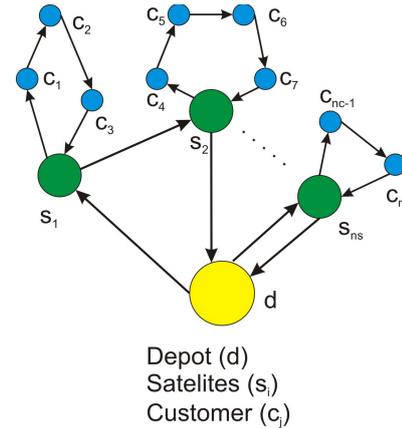


Fig. 6 Example of 2E-CVRP transportation network

The transformation of this model in the hybrid approach focused on the resizing of $Y_{k,i,j}$ decision variable by introducing additional imaginary volume of freight shipped from the satellite and re-delivered to it. Such transformation resulted in two facts. First of all, it forced the vehicle to return to the satellite from which it started its trip. Secondly, it reduced decision variable $Y_{k,i,j}$ to variable $Y_{i,j}$ which decreased the size of the combinatorial problem.

VI. NUMERICAL EXPERIMENTS

A. Cost optimization of supply chain with multimodal transport

In order to verify and evaluate the proposed approach, many numerical experiments were performed. All the examples relate to the supply chain with two manufacturers

TABLE IV
SUMMARY INDICES, PARAMETERS AND DECISION VARIABLES

Symbol	Description
<i>Indices</i>	
n_s	Number of satellites
n_c	Number of customers
$V_0 = \{v_0\}$	Depot
$V_s = \{v_{s1}, \dots, v_{sns}\}$	Set of satellites
$V_c = \{v_{c1}, \dots, v_{cnc}\}$	Set of customers
<i>Input parameters</i>	
m_1	Number of the 1st-level satellites
M_2	Number of the 2nd-level satellites
k_1	Capacity of the vehicles for the 1st level
k_2	Capacity of the vehicles for the 2nd level
d_i	Demand required by customer i
c_{ij}	Cost of the arc (i, j)
s_k	Cost of loading/unloading operations of a unit of freight in satellite k
<i>Decision variables</i>	
$X_{i,j}$	Is an integer variable of the 1st-level routing and is equal to the number of 1st-level vehicles using arc (i, j) .
$Y_{k,i,j}$	Is a binary variable of the 2nd-level routing and is equal to 1 if a 2nd-level vehicle makes a route starting from satellite k and goes from node i to node j and 0 otherwise
$Q1_{i,j}$	freight flow arc ij for the 1st-level
$Q2_{k,i,j}$	freight arc ij where k represents the satellite where the freight is passing through.
$z_{k,j}$	Binary variable that is equal to 1 if the freight to be delivered to customer j is consolidated in satellite k and 0 otherwise

($i=1..2$), three distributors ($s=1..3$), five customers ($j=1..5$), three modes of transport ($d=1..3$), and ten types of products ($k=1..10$). Other parameter values are shown in Appendix A1 [17].

The first series of experiments was designed to show the advantages of the hybrid approach used.

The experiments began with six examples: E1, E2, E3, E4, E5, E6 for the problem formulated in MILP (V) [17]. Two approaches were used to implement the proposed model: mathematical programming (LINGO) and the hybrid approach (LINGO, Eclipse, transformation). The examples E1 .. E6 varied in terms of the number of orders (No). The set of all orders for calculation examples are given in Appendix A.

The experiments were conducted to optimize examples E7, E8, which are implementations of the hybrid model (with logical constraints) in the hybrid approach.

The implementation of logic constraints for the hybrid model was as follows: product $k = 5$ cannot be distributed with product $k = 6$; product $k = 2$ cannot be distributed with product $k = 8$, and these products cannot be produced together. The results in the form of the objective function, the computation time, the number of discrete decision variables and constraints are shown in Table V.

TABLE V
THE RESULTS OF NUMERICAL EXAMPLES FOR BOTH APPROACHES

E(No)	MILP-LINGO				MILP-Hybrid			
	Fc	T	V	C	Fc	T	V	C
E1(5)	6680	7	1389	1351	6680	2	117	172
E2(10)	20439	28	1389	1621	20439	3	173	172
E3(15)	29107	55	1389	1891	29107	9	245	172
E4(20)	45710*	600**	1389	2161	45654	18	301	172
E5(25)	46660*	600**	1389	2431	46150	235	376	172
E6(30)	48946*	600**	1389	2701	48006	375	429	172
P(No)	Hybrid-Hybrid							
	Fc	T	V	C				
E7(10)	21143	194	193	202				
E8(20)	46069	366	321	202				
Fc	the optimal value of the objective function							
T	Solution finding time							
V/C	the number of integer variables/constraints							
*	the feasible value of the objective function after the time T							
**	calculation was stopped after 600s							

The analysis of the outcome indicates that the hybrid approach provided better results in terms of the time needed to find the solution in each case, and to obtain the optimal solution in some cases, which was impossible to do within the acceptable time limits using the traditional approaches.

B. Two-Echelon Capacitated Vehicle Routing Problem

For the final validation of the proposed hybrid approach, the benchmark (2E-CVRP) was selected. 2E-CVRP, a well described and widely discussed problem, corresponded to the issues to which our hybrid approach was applied.

The instances for computational examples were built from the existing instances for CVRP [24] denoted as E-n13-k4. All the instance sets can be downloaded from the website [25]. The instance set was composed of 5 small-sized instances with 1 depot, 12 customers and 2 satellites. The full instance consisted of 66 small-sized instances because the two satellites were placed over twelve customers in all 66 possible ways (number of combinations: 2 out of 12).

All the instances had the same position for depot and customers, whose coordinates were the same as those of instance E-n13-k4. Small-sized instances differed in the choice of two customers who were also satellites (En13-k4-2 (1,3), En13-k4-6 (1,6), En13-k4-61 (9,10) etc.).

The analysis of the results for the benchmark instances demonstrates that the hybrid approach may be a superior approach to the classical mathematical programming. For all examples, the solutions were found 2-16 times faster than they are in the classical approach.

As the presented benchmark was formulated as a MILP problem, the HSF was tested for the solution efficiency. Owing to the hybrid approach the 2E-CVRP models can be extended over logical, nonlinear, and other constraints.

TABLE VI
THE RESULTS OF NUMERICAL EXAMPLES FOR BOTH APPROACHES

E-n13-k4	MILP-LINGO				MILP-Hybrid			
	Fc	T	V	C	Fc	T	V	C
En13-k4-2	286	40371	368	1262	286	8720	186	1024
En13-k4-6	230	125	368	1262	230	55	186	1024
En13-k4-9	244	153	368	1262	244	44	186	1024
En13-k4-20	276	535	368	1262	276	32	186	1024
En13-k4-61	338	6648	368	1262	338	407	186	1024
Fc	the optimal value of the objective function							
T	time of finding solution							
V/C	the number of integer variables/constraints							
*	the feasible value of the objective function after the time T							

VII. CONCLUSION AND DISCUSSION ON POSSIBLE EXTENSION

The efficiency of the proposed approach is based on the reduction of the combinatorial problem and using the best properties of both environments. The hybrid approach (Table V, Table VI) makes it possible to find solutions better solutions in the shorter time.

In addition to solving larger problems faster, the proposed approach provides virtually unlimited modeling options.

Therefore, the proposed solution is recommended for decision-making problems in the supply chain that has a similar structure to the presented model (V). This structure is characterized by the constraints and objective function in which the decision variables are added together. Further work will focus on running the optimization models with non-linear and logical constraints, multi-objective, uncertainty etc. in the hybrid optimization framework.

References

- [1] Kanyalkar, A.P., Adil, G.K., *An integrated aggregate and detailed planning in a multi-site production environment using linear programming*. International Journal of Production Research 43, 2005, pp. 4431–4454.
- [2] Perea-lopez, E., Ydstie, B.E., Grossmann, I.E., *A model predictive control strategy for supply chain optimization*. Computers and Chemical Engineering 27, 2003, pp. 1201–1218.
- [3] Christian Lang J. *Production and Operations Management: Models and Algorithms*. Production and Inventory Management with Substitutions, Lecture Notes in Economics and Mathematical Systems Volume 636, 2010, pp 9–79.
- [4] Dang Quang, Nielsen Izabela, Steger-Jensen Kenn, Madsen Ole, *Scheduling a Single Mobile Robot for Part-Feeding Tasks of Production Lines*, Journal of Intelligent Manufacturing, 2013, DOI 10.1007/s10845-013-0729-y.
- [5] Apt K., Wallace M., *Constraint Logic Programming using Eclipse*, Cambridge University Press, 2006.
- [6] Sitek P., Wikarek J., *A Declarative Framework for Constrained Search Problems*, New Frontiers in Applied Artificial Intelligence, Lecture Notes in Artificial Intelligence, Nguyen, NT., et al. (Eds.), Vol. 5027, Springer-Verlag, Berlin-Heidelberg, 2008, pp. 728-737.
- [7] Bocewicz G., Wójcik R., Banaszak Z., *AGVs distributed control subject to imprecise operation times*, In: Agent and Multi-Agent Systems: Technologies and Applications, Lecture Notes in Artificial Intelligence, LNAI, Springer-Verlag, Vol. 4953, 2008, pp. 421–430.
- [8] Sitek P., Zaborowski M. *Grouping products in a follow-up production control system for parallel partitioned flow production lines*, Intelligent manufacturing systems IMS 2001: 6th IFAC Workshop, Pergamon, 2001, New York, pp.122-126.

- [9] Sitek P. Wikarek J. *The concept of decision support system structures for the distribution center*, MPER (Management and Production Engineering Review), vol.1, no.3, 2010, pp.63-69
- [10] Sitek P., Wikarek J., *Cost optimization of supply chain with multimodal transport*, Federated Conference on Computer Science and Information Systems (FedCSIS), 2012, pp. 1111–1118.
- [11] Sitek P., Wikarek J., *Supply chain optimization based on a MILP model from the perspective of a logistics provider*, Management and Production Engineering Review, 2012, pp. 49–61.
- [12] Sitek P., Wikarek J., *The Declarative Framework Approach to Decision Support for Constrained Search Problems*, INTECH, 2011, pp 163–182.
- [13] Jain V., Grossmann I.E., *Algorithms for hybrid MILP/CP models for a class of optimization problems*, INFORMS Journal on Computing 13(4), 2001, pp. 258–276.
- [14] Milano M., Wallace M., *Integrating Operations Research in Constraint Programming*, Annals of Operations Research vol. 175 issue 1, 2010, pp. 37 – 76.
- [15] Achterberg T., Berthold T., Koch T., Wolter K., *Constraint Integer Programming: A New Approach to Integrate CP and MIP*, Lecture Notes in Computer Science, Volume 5015, 2008, pp. 6–20.
- [16] Bockmayr A., Kasper T., *A Framework for Combining CP and IP, Branch-and-Infer, Constraint and Integer Programming Operations Research/Computer Science Interfaces Series, Volume 27, 2004, pp. 59-87.*
- [17] Sitek P., Wikarek J., *A hybrid approach to supply chain modeling and optimization*, Federated Conference on Computer Science and Information Systems (FedCSIS), 2013, pp. 1223–1230.
- [18] Perboli G., Tadei R., Vigo D., *The Two-Echelon Capacitated Vehicle Routing Problem: Models and Math-Based Heuristics*, Transportation Science, 2011, v45, pp.364-380.
- [19] Crainic, T., Ricciardi, N., Storchi, G., 2004. *Advanced freight transportation systems for congested urban areas*. Transportation Research part C 12, 119–137.
- [20] Schrijver A., *Theory of Linear and Integer Programming*, ISBN 0-471-98232-6, John Wiley & sons, 1998.
- [21] www.eclipse.org
- [22] www.lindo.com
- [23] Ricciardi, N., Tadei, R., Grosso, A., *Optimal facility location with random throughput costs*. Computers and Operations Research 29 (6), 2002,593–607.
- [24] Christofides, N., Elion, S., *An algorithms for the vehicle dispatching problem*. Operational Research Quarterly 20, 1969, pp.309–318.
- [25] <http://www.orgroup.polito.it/>
- [26] Rossi F., Van Beek P., Walsh T., *Handbook of Constraint Programming (Foundations of Artificial Intelligence)*, Elsevier Science Inc. New York, NY, USA © 2006.

APPENDIX A

TABLE I

THE SET OF ORDERS FOR COMPUTATIONAL EXAMPLES E1-E8

Name	k	j	T _{kj}	Z _{j,k}	Name	k	j	T _{kj}	Z _{j,k}
z 01	p1	m1	8	10	z 11	p1	m3	8	15
z 02	p2	m2	12	10	z 12	p2	m4	12	20
z 03	p3	m3	10	25	z 13	p3	m5	10	25
z 04	p4	m4	8	30	z 14	p4	m1	8	30
z 05	p5	m5	12	10	z 15	p5	m2	12	30
z 06	p6	m1	8	15	z 16	p6	m3	8	15
z 07	p7	m2	12	20	z 17	p7	m4	12	20
z 08	p8	m3	10	25	z 18	p8	m5	10	25
z 09	p9	m4	8	30	z 19	p9	m1	8	30
z 10	p10	m5	12	30	z 20	p10	m2	12	35
z 21	p1	m5	8	2	z 26	p6	m5	8	3
z 22	p2	m1	12	1	z 27	p7	m3	12	2
z 23	p3	m4	10	2	z 28	p8	m4	10	2
z 24	p4	m5	8	1	z 29	p9	m2	8	2
z 25	p5	m3	12	1	z 30	p10	m1	12	2