

A declarative decision support framework for scheduling groups of orders

Jarosław Wikarek

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

Krzysztof Bzdrya

Koszalin University of Technology,
Department of Computer Science and
Management, Poland e-mail:
krzysztof.bzdrya@tu.koszalin.pl

Abstract—This paper deals with declarative decision support framework for scheduling groups of orders. All orders in a group should be delivered at the same time after processing. The authors present a novel declarative approach to modeling and solving scheduling problems as a declarative decision support framework. The proposed framework makes it possible to ask different types of questions (general, specific, logical, etc.). It also allows, scheduling emerging orders or groups of orders without changing the existing schedules. To implement was used CLP (Constraint Logic Programming) environment. To increase the efficiency of the framework, particularly in the area of optimization made its integration with MP (Mathematical Programming) environment. The paper also presents the implementation of illustrative model, using the proposed framework. In addition, an efficiency analysis of the presented solution in relation to the application of mathematical programming have been conducted.

I. INTRODUCTION

THE proposed research problem (scheduling groups of orders) finds many applications in industrial and services companies, including but not limited to food, textile production industries, distributions, ceramic tile, supply chain, manufacturing of complex devices, fast-foods, restaurants, postal and courier services etc. Assume that each customer has different orders. Each order has a different process function and set of 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. In this type scheduling problems, in addition to standard constraints such as precedence or disjunction, new constraints appear related to the given group, concurrent delivery date, etc. In practice also logical constraints may occur, resulting from business, marketing or legal conditions. Therefore the modeling and solving of various constraints in scheduling groups of orders is a key issue. Managers/Decision-makers need to have schedules with defined parameters and/or the knowledge whether the schedule meets the requirements, which may be formulated as simple questions. Good environments for the modeling of constraints, questions and logi-

cal conditions include declarative environments, CLP (Constraint Logic Programming) in particular.

Our motivation was to develop an environment for the modeling and decision support of the problem for scheduling groups of orders. The use of this framework would help obtain quick answers to key questions (Is it possible...?, What If...?, What is the minimum/maximum..?) asked by managers/decision makers.

This paper proposes the concept of a declarative decision support framework for scheduling groups of orders and presents its implementation in the CLP environment. The illustrative example shows the potential of the framework.

The remainder of the article is organized as follows. Section 2 presents problem statement, research methodology, contribution etc. The concept and implementation aspects of a declarative decision support framework are provided in Section 3. Computational examples, tests of the implementation framework and discussion are presented in Section 4. Possible extensions of the proposed approach as well as the conclusions are included in Section 5.

II. PROBLEM STATEMENT AND METHODOLOGY

Scheduling methods for optimal and simultaneous service to groups of orders are proposed most often in the flexible flow-shop system (FFS). In the FFS system, processing is divided into several stages with parallel resources/machines at least in one stage. All of the orders should pass through all stages in the same order [1,2]. The objectives of the problem [2] are minimizing the total amount of time required to complete a group of orders and minimizing the sum of differences between the completion time of a particular order in the group and the delivery time of this group containing that order (waiting period). In practical applications, flexible flow-shop system is insufficient since the sequence of operations/tasks in the orders from different users is rarely the same.

The majority of models for scheduling of group orders presented in the literature refer to a single problem and optimization according to single criterion. Fewer studies are devoted to multiple-criteria [2]. Most of them are modeled and solved by operations research (OR) methods. Declarative environments such as CLP facilitate problem

modeling and introduction of logical and symbolic constraints [3,4,5]. Unfortunately, high complexity and the multiple types of constraints of decision-making models as well as combinatorial nature contribute to poor efficiency of modeling in OR methods and inefficient optimization in CLP. Therefore, a new approach to modeling and solving such problems was developed [6,7,8]. A declarative environment was chosen as the best structure for this approach especially in modeling [3,5,9,10]. Mathematical programming environment was used for problem optimization [11]. This integrated approach is the basis for the creation of the implementation environment to support managers. In addition to optimizing particular decision making problems connected with groups of orders, such environment allows asking various questions while processing the orders.

A. Problem description –illustrative example

This problem can be stated as follows. Orders Z_i for different group of product p enter the system in groups at different periods v . Each order consists of a set of operations and should be processed with specific set of parallel resources. It is assumed that there are no gaps between the operations of the order. The orders in each group Z_i should be delivered at the same time. Special points a at which orders are submitted and then delivered are introduced. The problem does not cover the configuration of the points but relates to handling orders, as many orders may come from one customer. Each order may be processed by a different resource set in any order.

In this case, decision support is to respond to the questions asked, which in general can be: specific questions, general questions, logic questions etc.

Possible questions (Q) for such problem are (including but not limited to):

- What is the minimum makespan for groups of orders Z_1, \dots, Z_n entering in period v_1, \dots, v_n at the point a_1, \dots, a_n ? (Q1)
- Is it possible to execute the new group of orders Z_{n+1}, \dots, Z_m from the period v_k with existing resources at specified period T ? (No change orders that are in progress.) (Q2)
- Is it possible to execute group of orders Z_1, \dots, Z_n in time T with use of the resource $k=N$? (Q3)
- What is the minimum use of resource k to execute orders Z_1, \dots, Z_n in time T ? (Q4)
- Is it possible to execute the new group of orders Z_{n+1}, \dots, Z_m from the period v_k with existing resources at specified period T and the use of resource $k=N$? (Q5)
- What is the minimum makespan for groups of orders/tasks Z_1, \dots, Z_n , entering in period v_1, \dots, v_n at the point a_1, \dots, a_n ? (with all the resources k reduced by $C\%$) (Q6)
- Is it possible to execute the groups of orders Z_1, \dots, Z_n in time T entering in period v_1, \dots, v_n at the point a_1, \dots, a_n ? with exclusively use resources k_i and k_j ? (Q7)

Decision variables of this problem are shown in Table I.

TABLE I.
DECISION VARIABLES

<i>Decision variables</i>	
Calculated number of periods g delivery of all orders for point a .	Tk_{p_a}
If at a given point a ordered product p then $Xz_{k_{a,p}}=I$, otherwise $Xz_{k_{a,p}}=0$	$Xz_{k_{a,p}}$
Number of period g in which operation o can be started for product p ordered at point a	$B_{a,p,o}$
If the execution of operation o for product p ordered at point a uses resource k in period g then $X_{a,p,o,k,g}=I$, otherwise $X_{a,p,o,k,g}=0$	$X_{a,p,o,k,g}$
If the execution of operation o for product p ordered at point a uses resource k in period g then $X_{O_{a,p,o,k,g}}=zk_{a,p}$, otherwise $X_{O_{a,p,o,k,g}}=0$	$X_{O_{a,p,o,k,g}}$
If g is the last period in which resource k is used in the execution of operation o for product p at point a then $Y_{a,p,o,k,g}=I$, otherwise $Y_{a,p,o,k,g}=0$	$Y_{a,p,o,k,g}$
If g is the last period in which orders are executed for point a then $W_{a,g}=I$, otherwise $W_{a,g}=0$	$W_{a,g}$
Number of period g from resource k can be used for operation o of product p ordered at point a	$S_{a,p,o,k}$

The set of reference constraints for the problem was created and its mathematical/formal notation is included in Appendix A.

Constraint (1) determines whether in a given point a product p has been ordered (setting the value of variable $Xz_{k_{a,p}}$). Constraints (2) ensures the order execution of operations for the product p (precedence constraint). Constraint (3) specifies the moment (period) from which resource k is needed to execute product p . Constraint (4) states no start is possible before orders appear. The term of delivery to the point a defines the constraint (5). Constraint (6) ensures that the number of available resources k in period g is not exceeded. Constraint (7) provides resource occupancy for the time of the order execution. Operations are not interrupted during their execution (8). Simultaneous completion of orders for product p from the given point a is ensured by constraints (9,10). Constraint (11) is responsible for the binarity of selected decision variables.

III. A DECLARATIVE DECISION SUPPORT FRAMEWORK FOR SCHEDULING GROUPS OF ORDERS-CONCEPT AND IMPLEMENTATION

The declarative decision support framework was proposed for scheduling groups of orders. The concept is based on the declarative programming paradigm, which allows high level programming with the use of predicates and facts. Due to the character of problems in the scheduling of groups of orders, CLP (Constraint Logic Programming) was selected from among many declarative options. The implementation of the framework was performed with the use of ECL^{PS} [12].

The following general assumptions were applied:

- possibility of modeling constraints of any type;
- automatic generation of implementation models in the form of MILP models;
- data recorded as facts;

- problem dynamic taken into account (possibility of introducing new orders or groups of orders).

Figure 1 presents the general concept of the framework. The framework comprises several phases: modeling, presolving, generating and solving. It has two inputs and uses the set of facts. Inputs are the set of questions and the set of constraints to the reference model of a given problem. Based on them, the primary model of the problem is generated as a CLP model, which is then presolved. The built-in CLP method (constraint propagation [13]) and the method of problem transformation designed by the authors [6,9] (Section 3A) are used for this purpose. Presolving procedure results on the transformed model CLP^T . This model is the basis for the automatic generation of the MILP (Mixed Integer Linear Programming) model, which is solved in MP (with the use of an external Solver or CLP as a library). The general concept of the framework consists in modeling and presolving of a problem in the CLP environment with the final solution (including optimization) found in the MP environment. This approach is the result of experience as well as extensive research devoted to both environments [6,9,10] and their integration[6,14]. In all its phases, the framework uses the set of facts having the structure appropriate for the problem being modeled and solved (Fig. 2). The set of facts is the informational layer of the framework, which can be implemented as a relational database, XML database, etc.

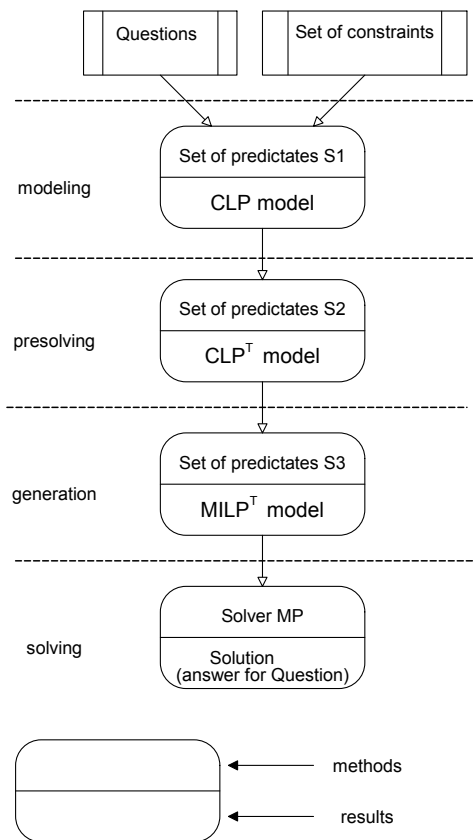


Fig. 1 A concept of a declarative decision support framework

In order to take into account the dynamics resulting, for example, from new orders, the $MILP^T$ model is solved iterative way with the use of the algorithm [15]. The main characteristic of this algorithm is iterative activation of MP solver and update of resource availability [15].

A. Presolving

The presolving phase is an important element of the framework as it makes it possible to simplify the model for the problem being solved and to reduce the combinatorial search space.

For the presolving phase to be effective, unfeasible combinations of model dimensions have to occur. In practice, unfeasible combinations of the index of decision variables and/or facts occur.

The proposed framework uses constraint propagation and transformation for the presolving procedure. Constraint propagation is a concept and method that appears in constrained-based environments. Constraint propagation embeds any reasoning which consists in explicitly forbidding values from some variable domain of a problem, because all constraints can not be satisfied otherwise. Transformation transforms decision variables of the problem along with constraints and facts. The transformation method for the illustrative example is shown in Fig. 2, and the post-transformation variables are compiled in Table AII. For the problem presented, the transformation consisted in the change from the problem’s operational representation into the resource representation. This resulted in the removal of all decision variables, parameters, etc. From the operation index, thereby reducing their numbers. The new set of decision variable, constraints and facts was the basis for creating the CLP^T model.

IV. COMPUTATION EXAMPLES FOR ILLUSTRATIVE MODEL

In order to verify and evaluate the proposed framework, many numerical experiments were performed for the illustrative example. In the first phase, all the experiments relate to the system with five points ($a=1..5$), eight order types (products) ($p=1..8$), eight resource types ($k=1..8$), thirty time periods ($g=1..30$) and eleven orders $z_{g,v,p}$ (five groups of orders Z_i in three periods)

All data instances for these experiments were recorded in the form of facts and included Appendix B.

Computational experiments consisted in asking questions Q1..Q7 to illustrative example. For each question was generated and solved suitable implementation model using declarative decision support framework. Orders are placed in groups for $v_1=1$, $v_2=2$ and $v_3=5$ (only for Q2 and Q5) periods. The answers to these questions are shown in Table II. Figure 3 shows the implementation schedule of all group of orders for question Q1 (minimizing makespan). A proper schedule utilization of resources corresponding to the schedule of Figure 3 is shown in Figure 4. By contrast, Figure 5 shows the implementation schedule of all group of orders for question Q6 (with all the resources k reduced by

50%). In analogy to the previous question, a proper schedule utilization of resources corresponding to the schedule of Figure 5 is shown in Figure 6.

The answer to the question Q4 determines the minimum resource requirements ($k_1..k_8$) necessary to complete all group of orders within T (Table II).

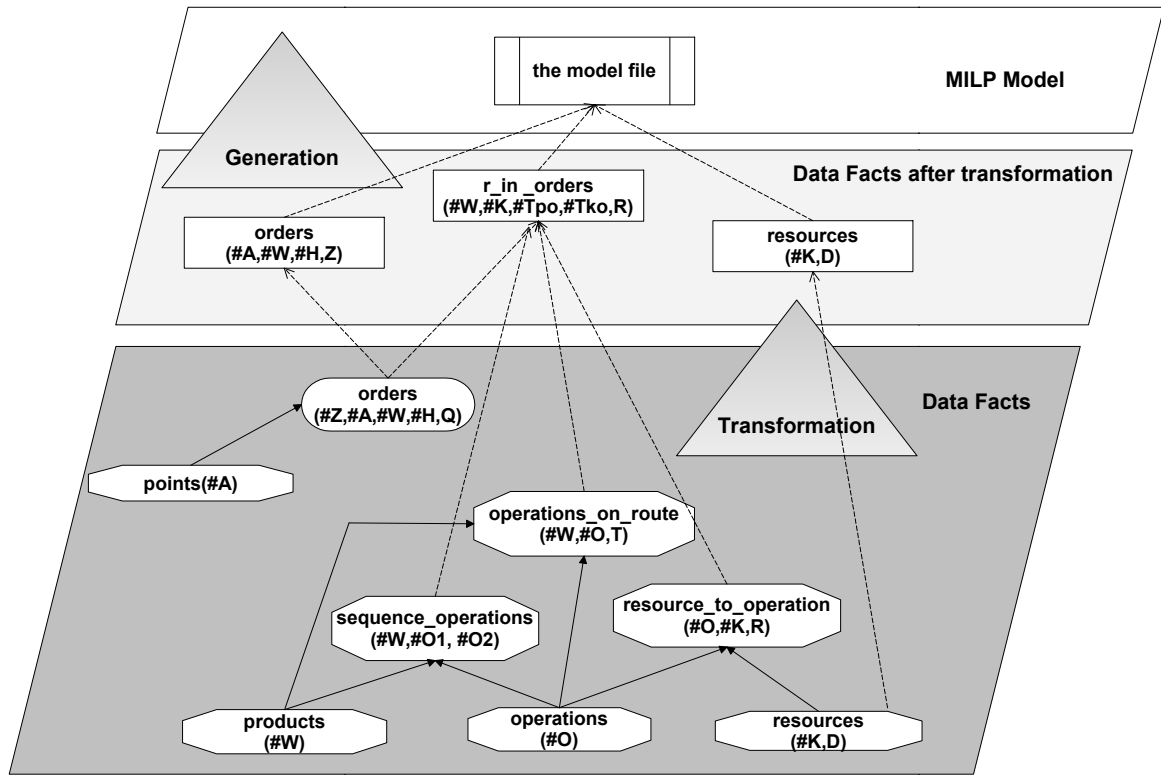


Fig. 2 Information layer of the framework (data fact before and after transformation, MILP model file)

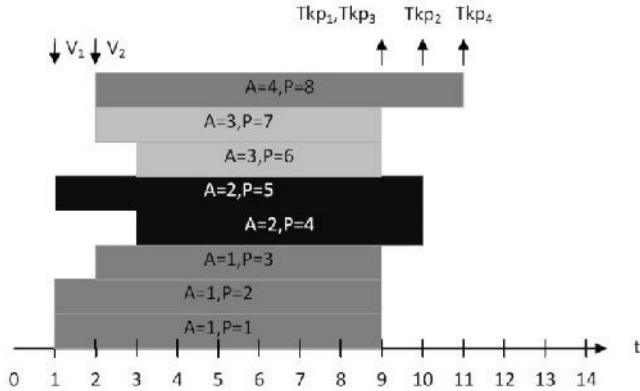


Fig. 3 Gantt chart for illustrative question Q1 ($V_1=1, V_2=2, Tkp_1=Tkp_3=9, Tkp_2=10, Tkp_4=11, C_{max}=11$)

The answers to the remaining questions confirm feasibility or unfeasibility of the schedule in the set conditions and at defined parameters (Table II). Information about the feasibility of the schedule and about available resources is especially useful when a new group of orders (Q2,Q5) appears. The last question from the set (Q7) is the question with a logical condition relating to disjoint use of resources.

In the second phase of the experiments, a comparative analysis was performed for questions Q1 and Q6 (most compute-intensive of all) in two environments (declarative decision support framework and MP) to evaluate the

effectiveness and efficiency of the proposed framework relative to the classical MP environment.

Obtained more than 400-fold reduction of time searching for solutions (Table III). This is due to the fact that the application of the framework has allowed the reduction of decision variables from 134616 to 13873 (10-fold) and constraints from 252541 to 31239 (8-fold) (Table III).

TABLE II. ANSWERS TO QUESTION FOR ILLUSTRATIVE EXAMPLE

Question	Parameters	Answer
Q1	---	$C_{max}=11$
Q2 _A	$T=12$	NO
Q2 _B	$T=14$	YES
Q3 _A	$T=12, k_1=k_2..=k_8=k=12$	NO
Q3 _B	$T=13, k_1=k_2..=k_8=k=12$	YES
Q4	$T=20$	$k_1=8, k_2=8, k_3=6, k_4=7, k_5=4, k_6=5, k_7=1, k_8=6,$
Q5 _A	$Z_3, v_3=5, k=12, T=12$	NO
Q5 _B	$Z_3, v_3=5, k=12, T=13$	YES
Q6 _A	15%	$C_{max}=11$
Q6 _B	50%	$C_{max}=12$
Q7 _A	$T=20, k_5 \text{ i } k_7$	YES
Q7 _B	$T=20, k_5 \text{ i } k_8$	NO

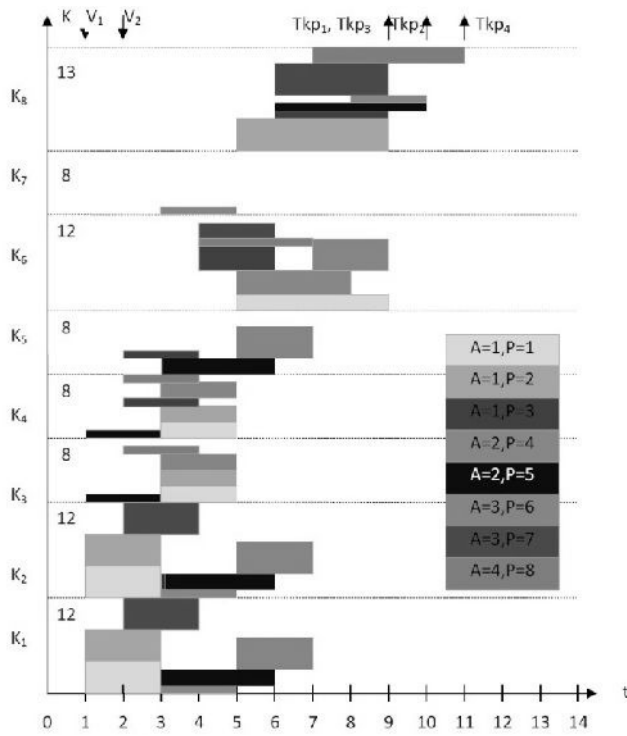


Fig. 4 Gantt chart for illustrative question Q1

TABLE III.

NUMERICAL EXPERIMENTS ON THE EFFICIENCY

Model	V(Vint)	C	Answer	T
Q1				
MILP	134616(134435)	252541	11	867
MILP ^T	13873 (13690)	31239	11	2
Q6 (50%)				
MILP	134616(134435)	252541	12	856
MILP ^T	13873 (13690)	31239	12	2

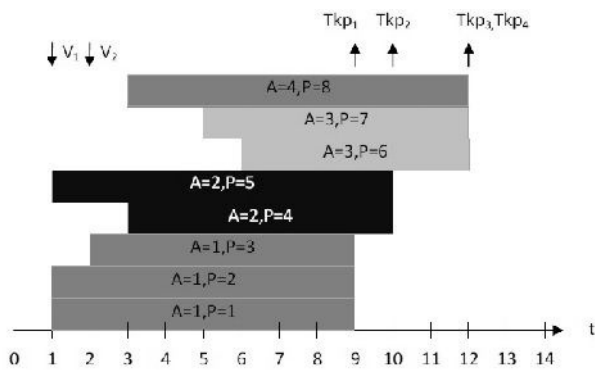


Fig. 5 Gantt chart for illustrative question Q6B ($V_1=1, V_2=2, Tkp_1=9, Tkp_2=10, Tkp_3=Tkp_4=12, C_{max}=12$)

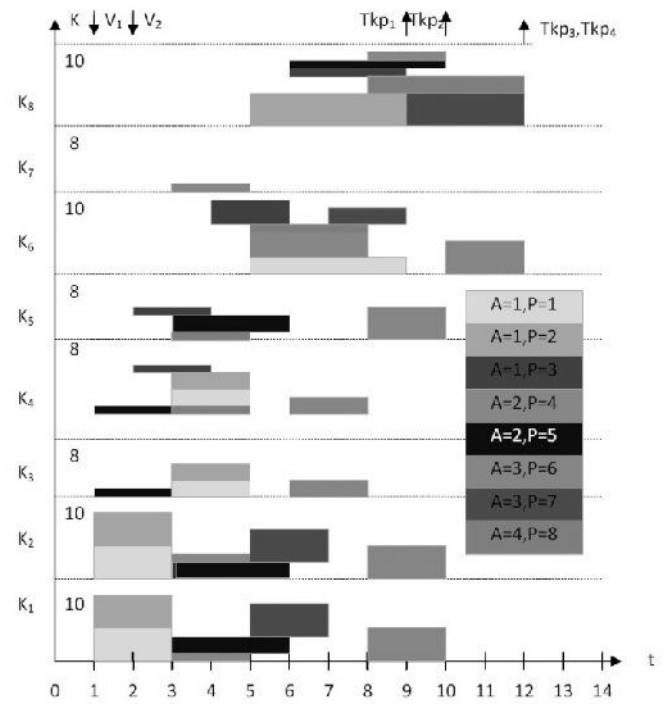


Fig. 6 Gantt chart for illustrative question Q6B

TABLE IV.
STRUCTURE OF THE FACTS

Fact	Description
points (#A)	points of ordering and delivery A – point id
products (#W)	products, services, etc. W – product id
operations (#O).	technological operations, tasks, etc. O – operation id
resources (#K,D).	resources (machines, tools, peoples, etc.) K – resource id, D - the number of available resources
resource_to_operation (#O,#K,R)	the resources necessary to execute the operations O – operation id, K – resource id, R – the number of resource k needed for the operation
operations_on_route (#W,#O,T)	a set of operations for the product (route), W – product id, O – operation id, T – duration of the operation
sequence_operations (#W,#O,#O)	the order of operations for the product, W – product id, O – operation id
orders (#Z,#A,#W,V,Q)	orders, Z – orders id, A – point id, W – product id, V – order period, Q – order size
r_in_orders (#W,#K,#Tpo,#Tko,R)	the allocation of resources to products, W – product id, K – resource id, Tpo – the beginning of the use of the resource, Tko – the end of the use of the resource, R – the number of resource k needed for the operation

V. CONCLUSIONS

Two types of questions can be asked in the proposed declarative decision support framework.

General questions may require domain solution, which in practice determines the availability of resources to execute orders, timely execution etc. The specific wh-questions will in practice define the best, fastest, cheapest, or the most

expensive of the possible solutions. To obtain answers to these questions, optimization is necessary.

Both question types can contain logical conditions relating, for example, to the disjoint use of resources, etc.

The illustrative example shows only part of potential of the framework designed to increase both the speed and the size of the problems solved.

This is particularly evident if we compare the possibilities of the framework in relation to the classical approach based on mathematical programming (Table III).

Further work will consist in the implementation of more complex models, uncertainty, product demand interdependencies [16], fuzzy logic [17,18] etc., and as a cloud internet application [19]. New questions will be implemented to broaden the scope of decision support.

APPENDIX A

TABLE AI.
SUMMARY INDICES, PARAMETERS

Sets	
Set of points (tables)	LA
Set of resources	LK
Number of periods	LG
Number of periods in which orders can be entered	LV
The set of operations	LO
The set of products (orders)	LP
Indices	
Points (tables)	a=1..LA
Resources	k=1..LK
Period	g=1..LG
Period in which orders can be entered	v=1..LV
Operation	o=1..LO
Products	p=1..LP
Parameters	
Number used to convert periods to moments (for connecting index g with variable)	pp _g
The number of available resources k in the period g	dkg _{k,g}
Duration of operation o for product p	t _{p,o}
Time to complete all operations o for the product p	t1 _{p,o}
If the operation o_1 precedes o_2 for product p than kol _{p,o1,o2} =1 otherwise kol _{p,o1,o2} =0	kol _{p,o1,o2}
If the operation o uses resource k than zas _{o,k} =1 otherwise zas _{o,k} =0	zas _{o,k}
Number of k resources needed for execution operation o	r _{o,k}
How much of the product p ordered at the point a .	Zk _{a,p}
The number of period in which new orders appeared.	ts
Inputs	
How much of the product p ordered at the point a in the period v .	zgp _{v,a,p}

TABLE AII.
DECISION VARIABLES AFTER TRANSFORMATION

Decision variables	
Calculated number of periods g delivery of all orders for point a .	Tkp _a
If at a given point a ordered product p then $Xzk_{a,p}=1$, otherwise $Xzk_{a,p}=0$	Xzk _{a,p}
If the execution of product p ordered at point a uses resource k in period g then $X_{a,p,k,g}=1$, otherwise $X_{a,p,k,g}=0$	X _{a,p,k,g}
If the execution of product p ordered at point a uses resource k in period g then $Xo_{a,p,k,g}=zk_{a,p}$ otherwise $Xo_{a,p,k,g}=0$	Xo _{a,p,k,g}
If g is the last period in which resource k is used in the execution of product p at point a then $Y_{a,p,k,g}=1$, otherwise $Y_{a,p,k,g}=0$	Y _{a,p,k,g}
If g is the last period in which orders are executed for point a then $W_{a,g}=1$, otherwise $W_{a,g}=0$	W _{a,g}
Number of period g from resource k can be used for product p ordered at point a	S _{a,p,k}

$$Xzk_{a,p} \cdot LG \geq Zk_{a,p} \quad \forall a = 1..LA, p = 1..LP \quad (1)$$

$$Xzk_{a,p} \leq zk_{a,p} \quad \forall a = 1..LA, p = 1..LP$$

$$B_{a,p,o_1} + t_{p,o_1} \cdot Xzk_{a,p} = B_{a,p,o_2} \quad \forall a = 1..LA, p = 1..LP, o_1, o_2 = 1..LO: kol_{p,o_1,o_2} = 1 \quad (2)$$

$$S_{a,p,o,k} = B_{a,p,o} \quad \forall a = 1..LA, p = 1..LP, o = 1..LO, k = 1..LK: r_{o,k} > 0 \quad (3)$$

$$S_{a,p,o,k} = 0 \quad \forall a = 1..LA, p = 1..LP, o = 1..LO, k = 1..LK: r_{o,k} = 0$$

$$B_{a,p,o} \geq ts \cdot Xzk_{a,p} \quad \forall a = 1..LA, p = 1..LP, o = 1..LO: t_{p,o} > 0 \quad (4)$$

$$Tkp_a \geq B_{a,p,o} + (t1_{p,o} + t_{p,o}) \cdot Xzk_{a,p} \quad \forall a = 1..LA, p = 1..LP \quad (5)$$

$$\sum_{a=1}^{LA} \sum_{p=1}^{LP} \sum_{o=1}^{LO} (Xo_{a,p,o,k,g} \cdot r_{o,k}) \leq dkg_{k,g} \quad \forall k = 1..LK, g = 1..LG \quad (6)$$

$$\sum_{g=1}^{LG} X_{a,p,o,k,g} = t_{p,o} \cdot Xzk_{a,p}$$

$$\forall a = 1..LA, p = 1..LP, o = 1..LO, k = 1..LK \quad (7)$$

$$\sum_{g=1}^{LG} Xo_{a,p,o,k,g} = t_{p,o} \cdot Zk_{a,p}$$

$$\forall a = 1..LA, p = 1..LP, o = 1..LO, k = 1..LK$$

$$X_{a,p,o,k,g-1} - X_{a,p,o,k,g} \leq Y_{a,p,o,k,g-1}$$

$$\forall a = 1..LA, p = 1..LP, o = 1..LO, k = 1..LK, g = 2..LG \quad (8)$$

$$\sum_{g=1}^{LG} Y_{a,p,o,k,g} \leq Xzk_{a,p}$$

$$\forall a = 1..LA, p = 1..LP, o = 1..LO, k = 1..LK$$

$$Y_{a,p,o,k1,g} = Y_{a,p,o,k2,g} \quad \forall a = 1..LA, p = 1..LP, o = 1..LO, k1, k2 = 1..K, g = 1..LG: zas_{o,k1} = 1 \wedge zas_{o,k2} = 1 \quad (9)$$

$$Tkp_a = \sum_{g=1}^{LG} pp_g \cdot W_{a,g} \quad \forall a = 1..LA$$

$$Y_{a,p,o,k,g-t1_{p,o}} \leq W_{a,g}$$

$$\forall a = 1..LA, p = 1..LP, o = 1..LO, k = 1..LK, g - t1_{p,o} \geq 0 \quad (10)$$

$$Y_{a,p,o,k,g} = W_{a,g+t1_{p,o}}$$

$$\forall a = p..LP, o = 1..LO, k = 1..LK, g = 1..LG + t1_{p,o} \leq LG$$

$$\begin{aligned}
& X_{a,p,o,k,g} \in \{0,1\} \\
\forall a = 1..LA, p = 1..LP, o = 1..LO, k = 1..LK, g = 1..LG \\
& X_{o,a,p,o,k,g} \in C \\
\forall a = 1..LA, p = 1..LP, o = 1..LO, k = 1..LK, g = 1..LG \\
& Y_{a,p,o,k,g} \in \{0,1\} \\
\forall a = 1..LA, p = 1..LP, o = 1..LO, k = 1..LK, g = 1..LG \quad (11) \\
& S_{a,p,o,k} \in C \forall a = 1..LA, p = 1..LP, o = 1..LO, k = 1..LK \\
& Tkp_a \in C \forall a = 1..LA \\
& B_{a,p,o} \in C \forall a = 1..LA, p = 1..LP, o = 1..LO \\
& W_{a,g} \in C \forall a = 1..LA, g = 1..LG \\
& Xzk_{a,p} \in \{0,1\} \forall a = 1..LA, p = 1..LP
\end{aligned}$$

APPENDIX B

```

points('A1'). points('A2'). points('A3').
points('A4'). points('A5').
products('W1'). products('W2'). products('W3').
products('W4'). products('W5'). products('W6').
products('W7'). products('W8').
operations('O1'). operations('O2').
operations('O3'). operations('O4').
operations('O5'). operations('O6').
operations('O7'). operations('O8').
operations('O9'). operations('O10').
resources('K1',20). resources('K2',20).
resources('K3',20). resources('K4',20).
resources('K5',20). resources('K6',20).
resources('K7',20). resources('K8',20).
resource_to_operation('O1','K1',2).
resource_to_operation('O1','K2',2).
resource_to_operation('O2','K3',1).
resource_to_operation('O2','K4',1).
resource_to_operation('O3','K1',2).
resource_to_operation('O3','K2',2).
resource_to_operation('O3','K5',2).
resource_to_operation('O4','K1',1).
resource_to_operation('O4','K2',1).
resource_to_operation('O4','K7',1).
resource_to_operation('O5','K4',1).
resource_to_operation('O5','K5',1).
resource_to_operation('O6','K6',3).
resource_to_operation('O7','K8',1).
resource_to_operation('O8','K6',2).
resource_to_operation('O9','K6',1).
resource_to_operation('O9','K6',1).
resource_to_operation('O10','K8',2).
operations_on_route('W1','O1',2).
operations_on_route('W1','O2',2).
operations_on_route('W1','O9',4).
operations_on_route('W2','O1',2).
operations_on_route('W2','O2',2).
operations_on_route('W2','O10',4).
operations_on_route('W3','O5',2).
operations_on_route('W3','O6',2).
operations_on_route('W3','O7',3).
operations_on_route('W4','O4',2).
operations_on_route('W4','O6',3).
operations_on_route('W4','O7',2).
operations_on_route('W5','O2',2).
operations_on_route('W5','O3',3).
operations_on_route('W5','O7',4).
operations_on_route('W6','O3',2).
operations_on_route('W6','O8',2).
operations_on_route('W7','O1',2).
operations_on_route('W7','O9',2).
operations_on_route('W7','O10',3).
operations_on_route('W8','O2',2).

```

```

operations_on_route('W8','O9',3).
operations_on_route('W8','O10',4).
sequence_operations('W1','O1','O2').
sequence_operations('W1','O2','O9').
sequence_operations('W2','O1','O2').
sequence_operations('W2','O2','O10').
sequence_operations('W3','O5','O6').
sequence_operations('W3','O6','O7').
sequence_operations('W4','O6','O6').
sequence_operations('W4','O6','O7').
sequence_operations('W5','O2','O3').
sequence_operations('W5','O3','O7').
sequence_operations('W6','O2','O3').
sequence_operations('W6','O3','O8').
sequence_operations('W7','O1','O9').
sequence_operations('W7','O9','O10').
sequence_operations('W8','O2','O9').
sequence_operations('W8','O9','O10').
orders('Z1','A1','W1',1,2).
orders('Z1','A1','W2',1,2).
orders('Z1','A1','W3',1,1).
orders('Z2','A2','W4',1,1).
orders('Z2','A2','W5',1,1).
orders('Z3','A3','W6',2,2).
orders('Z3','A3','W7',2,2).
orders('Z4','A4','W8',2,1).
orders('Z5','A5','W1',5,1).
orders('Z5','A5','W7',5,2).
orders('Z5','A5','W8',5,1).

```

REFERENCES

- [1] I. Ribas, R. Leisten, J.M. Framinan, “Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective“, in: *Computer Operation Research*, 37, pp.1439–1454, 2010.
- [2] B. Tadayon, N. Salmasi, “A two-criteria objective function flexible flowshop scheduling problem with machine eligibility constraint“, in: *The International Journal of Advanced Manufacturing Technology*, 64(5-8), pp. 1001-1015, 2013.
- [3] K. Apt, M. Wallace, “Constraint Logic Programming using Eclipse“, Cambridge: Cambridge University Press, 2006.
- [4] G. Bocewicz, I. Nielsen, Z. Banaszak, “Iterative multimodal processes scheduling“, in: *Annual Reviews in Control*, 38(1), pp. 113-132, 2014.
- [5] F. Rossi, P. Van Beek, T. Walsh, “Handbook of Constraint Programming“, New York: Elsevier Sc. Inc, 2006.
- [6] P. Sitek J. Wikarek, “A hybrid method for modeling and solving constrained search problems“, in: *Federated Conference on Computer Science and Information Systems (FedCSIS 2013)*, pp. 385-392, 2013.
- [7] P. Sitek, J. Wikarek, “A Hybrid Programming Framework for Modeling and Solving Constraint Satisfaction and Optimization Problems“, in: *Scientific Programming*, vol. 2016, Article ID 5102616, 2016. doi:10.1155/2016/5102616.
- [8] P. Sitek, J. Wikarek, “A hybrid framework for the modelling and optimisation of decision problems in sustainable supply chain management“, in: *International Journal of Production Research*, pp. 6611-6628, 2015. doi:10.1080/00207543.2015.1005762.
- [9] P. Sitek, “A hybrid CP/MP approach to supply chain modelling, optimization and analysis“, in: *Federated Conference on Computer Science and Information Systems (FedCSIS)*, pp. 1345-1352, 2014. doi:10.15439/2014F89
- [10] P. Sitek, “A hybrid approach to the two-echelon capacitated vehicle routing problem (2E-CVRP)“, in: *Advances in Intelligent Systems and Computing*, 267, pp. 251–263, 2104. doi:10.1007/978-3-319-05353-0_25.
- [11] A. Schrijver, “Theory of Linear and Integer Programming“, John Wiley & Sons, New York, NY, USA, 1998.
- [12] Eclipse, 2015, Eclipse - The Eclipse Foundation open source community website, Accessed August 12, www.eclipse.org.
- [13] B.M.W. Cheng, K.M.F. Choi, J.H.M. Lee, J.C.K. Wu, “Increasing Constraint Propagation by Redundant Modeling: an Experience Report“, *Constraints* May 1999, Volume 4, Issue2, pp 167-192, 1999.

- [14] M. Milano, M. Wallace M., "Integrating Operation Research in Constraint Programming", in *Annals of Operation Research*, 175(1), 2010, pp. 37-76, DOI:10.1007/s10479-009-0654-9.
- [15] P. Sitek, J. Wikarek, "A novel approach to decision support and optimization of group job handling for multimodal processes in manufacturing and services" in *15th IFAC/IEEE/IFIP/IFORS Symposium on Information Control Problems in Manufacturing*, Ottawa, Kanada, 2015, pp 2183-2188, doi:10.1016/j.ifacol.2015.06.401
- [16] P. Nielsen, I. Nielsen, K. Steger-Jensen, Analyzing and evaluating product demand interdependencies in *Computers in Industry*, 61 (9), 2010, 869-876, doi:10.1016/j.compind.2010.07.012.
- [17] M. Relich, W. Muszynski, The use of intelligent systems for planning and scheduling of product development projects in *Procedia Computer Science*, vol. 35, 2014, pp. 1586–1595.
- [18] G. Kłosowski, A. Gola, A. Świć, Application of Fuzzy Logic Controler for Machine Load Balancing in *Discrete Manufacturing System*, [in:]. K. Jackowski et. al. (Eds.): IDEAL 2015, LNCS 9375, 2015, pp. 256-263.
- [19] S. Bak, R. Czarnecki, S. Deniziak "Synthesis of Real-Time Cloud Applications for Internet of Things", in *Turkish Journal of Electrical Engineering & Computer Sciences*, 2013, DOI: 10.3906/elk-1302-178.