

A Hybrid Multi-Objective Programming Framework for Modeling and Optimization of Supply Chain Problems

Paweł Sitek

Institute of Management Control Systems, Kielce University of Technology Al. 1000-lecia PP 7, 25-314
Kielce, Poland,

Email: sitek@tu.kielce.pl

□

Abstract—This paper presents a hybrid programming framework for solving multi-objective optimization problems in supply chain. The proposed approach consists of the integration and hybridization of two modeling and solving environments, i.e., constraint logic programming and mathematical programming, to obtain a programming framework that offers significant advantages over the classical approach derived from operational research. The strongest points of both components are combined in the hybrid framework, which by introducing transformation allows a significant reduction in size of a problem and the optimal solution is found a lot faster. This is particularly important in the multi-objective optimization where problems have to be solved over and over again to find a set of Pareto-optimal solutions. An over two thousand-fold reduction in size was obtained for the illustrative examples together with a few hundred-fold reduction in the speed of finding the solution in relation to the mathematical programming method. In addition, the proposed framework allows the introduction of logical constraints that are difficult or impossible to model in operational research environments.

I. INTRODUCTION

SUPPLY chain (SC) is an integrated process in which a group of several organizations and/or companies, such as suppliers, producers, distributors and retailers, work together to acquire raw materials with a view to converting them into end products which they distribute to retailers [1]. Decision and optimization problems occurring in the real-world supply chain are characterized by multiple objectives, constraints and many different decision variables. The presence of multiple conflicting objectives and constraints is natural and results from the complexity and interrelated character of problems and different interests of individual supply chain participants. Environmental aspects such as CO₂ emissions, noise, etc., which are a new type of constraints in SC, are emerging to become an important factor in the design of supply chains. Operational research models, with mathematical programming (MP) in particular, are most often used. They include MILP (mixed integer linear programming), MIP (mixed integer programming), IP (integer programming), etc. and MOOP (multi-objective

optimization problem) [2]. The vast majority of these models have a very large number of constraints, and two major difficulties appear in their application. First, discrete optimization problems, both single and multi-objective, contain many discrete decision variables. This increases their computational complexity and the finding of the optimal solution is long and costly. Second, mathematical programming models have linear constraints, which is insufficient for the description of many of the SC problems. This paper deals with a problem of supply chain modeling, multi-objective optimization and solving. An important contribution of the presented approach is to propose a programming framework that supports the hybrid modeling, hybrid multi-objective optimization and analysis of decision problems in the supply chain. In this programming framework two environments are hybridized, constraint logic programming (CLP) and mathematical programming (MP), in which constraints are treated in different ways and different methods are implemented to use the strengths of both for solving complex and constrained problems. The hybrid approach offers a lot more possibilities and higher efficiency in both the modeling and multi-objective optimization. The concept of hybridization is complemented by the interaction algorithm and a complete transformation of the problem and together creating an application programming framework. The rest of the paper is organized as follows: Section 2 describes literature review. Next Section presents Methodology. Section 4 is about our motivation. Section 5 gives the concept of the novel constraint logic programming approach with MP-based solver and implementation platform. The optimization models as the illustrative examples are described in Section 6. Computational examples and tests of the implementation platform are presented in Section 7. The discussion on possible extensions of the proposed approach and conclusions is included in Section 8.

II. LITERATURE REVIEW OF SUPPLY CHAINS MODELS AND MULTI-OBJECTIVE OPTIMIZATION

The build-to-order supply chain (BOSC) model, a key operation model for providing services/products, has received more attention in recent years, car manufacturers

□ This work was not supported by any organization

including [3]. In the BOSC model, production activities are not executed until orders from customers have been received, which can effectively reduce the costs of demand prediction and inventory and credibly reflect market demands. When BOSC is started, the selection of suppliers becomes the priority. Product assembly begins after this selection. Based on the findings reported in the literature [4],[5]. BOSC is a successful supply chain model that is currently widely in use. There are two most widely encountered objectives of the objective function in multi-objective programming models for SC. The first objective is the cost of activity in the supply chain, including the cost of particular chain link, transport, work and even product design, etc. The second objective is associated with the costs or volumes of CO2 emission and other environmental aspects [6]. The second objective may also comprise delivery time [7] or, calculated in many different ways, the level of customer satisfaction [8],[9]. In [10] the multi-objective optimization mathematical model of BOSC has been presented. There are three objective functions. The first is cost minimizing including order cost, purchase cost and transport cost. The second is minimizing the maximum time of transporting the purchased parts to customers (delivery). The last is the part quality. Many researchers have recently reported the results of their multi-objective optimization studies. For example, a multi-objective programming model is proposed in [2],[11] to analyze solid waste management. The model for simultaneously optimizing the operations of both integrated logistics and its corresponding used-product reverse logistics in a close-looped supply chain has been presented in [12]. The common feature of these problems is the number of decision variables resulting from the allocation of resources, choice of location, route selection, choice of factory and distribution center, choice of mode of transport etc. These are usually binary and/or integer decision variables. Besides, all the problems are characterized by a large number of constraints binding decision variables. The overview of the models and algorithms of these problems is shown in [2],[27].

A. Multi-objective optimization

The multi-objective optimization problem (MOOP) can be defined as the problem of finding a vector of decision variables \hat{x} , which optimizes a vector of N objective functions $f_i(\hat{x})$ where $i = 1, 2, \dots, N$; subject to inequality constraints $g_j(\hat{x}) \geq 0$ and equality constraints $h_k(\hat{x}) = 0$ where $j = 1, 2, \dots, J$ and $k = 1, 2, \dots, K$.

A set of objective function is a multi-dimensional space, in addition to typically the decision space. This additional space is called the objective space Z . For each solution \hat{x} in the decision variable space, there exists a point in the objective space:

$$\hat{f}(\hat{x}) = Z(z_1, z_2, \dots, z_N)^T$$

In a MOOP, we want to find a set of values for the decision variables that optimizes a set of objective functions. A decision vector \hat{x} is said to dominate a decision vector \hat{y} (i.e. $\hat{x} > \hat{y}$) if:

$$f_i(\hat{x}) \leq f_i(\hat{y}) \quad \forall i \in \{1, 2, \dots, N\}$$

and

$$\exists i \in \{1, 2, \dots, N\} \mid f_i(\hat{x}) < f_i(\hat{y})$$

All decision vectors that are not dominated by any other decision vector are called non-dominated or Pareto-optimal and constitute the Pareto-optimal front/set. There are several methods for find the Pareto-optimal set of these optimization problems. Among the most widely techniques are: ϵ -constraint method, weighting method, goal programming, sequential optimization etc. [13].

III. METHODS AND METHODOLOGY

The key problem in the modeling and optimization of problems in the supply chain are multiple constraints of different types and character-linear, integer, non-linear, logical etc. Constraints are logical relations between variables, each variable taking a value from a specific domain. Thus a constraint restricts the possible values that a variable can take, i.e. it represents some partial information about the variables of interest. Constraints are:

- declarative, they specify a relationship between entities (decision variables) without determining a specific computational or programming procedure;
- additive, we are interested in the conjunction of constraints and not in the order in which they are imposed;
- rarely independent, normally constraints share decision variables.

Thus constraints are a natural medium and form to express problems in many fields, especially in logistic, transport, manufacturing, scheduling, distribution, supply chain etc. by all (researchers, practitioners, professionals, end-users etc.). In the above problems, there are resource, financial, capacity, time, transportation, environmental, multimodal, sale etc. constraints. Based on numerous studies and our own experience, the constraint-based environment [14], [15], [16], [17], [29] is believed to offer a very good framework for representing the knowledge, information and methods needed for the decision support and optimization. The central issue for a constraint-based environment is a constraint satisfaction problem (CSP) [14]. Constraint satisfaction problem is the mathematical problem defined as a set of elements whose state must satisfy a number of constraints. Constraint satisfaction problems (CSPs) 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 [15]. 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 (predicates). In CLP 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. Unfortunately, discrete optimization is not a strong suit of CP-based environments. This weakness is more pronounced in multi-objective optimization problems, where a parameterized problem of a single-objective optimization problem has to be solved multiple times depending on the size of Pareto set. Based on [14],[15] and previous work on hybridization [16],[17],[18] some advantages and disadvantages of these environments have been observed. The hybrid approach of constraint logic programming and mathematical programming can help to solve optimization problems that are intractable with either of the two methods alone [19],[20],[21]. In both MP and CLP, there is a group of constraints that can be solved with ease and a group of constraints that are difficult to solve. Both MP 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 [21]. MP 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.) [14]. Integrity constraints are difficult to solve using mathematical programming methods and often the real problems of MP make them NP-hard. In CP/CLP, domain constraints with integers are easy to solve. The system of such constraints can be solved over integer variables in polynomial time. The inequalities between many variables, general linear constraints, and symbolic constraints are difficult to solve, which makes real problems in CP/CLP NP-hard [21]. This type of constraints reduces the strength of constraint propagation. As a result, CP/CLP is incapable of finding even the first feasible solution [16].

IV. MOTIVATION AND CONTRIBUTION

The motivation and contribution behind this work was to apply a hybrid approach as a hybrid multi-objective programming framework for supply chain problems. The hybrid multi-objective programming framework is a concept that combines hybrid approach with iterative algorithm (Appendix A) in the context of multi-criteria optimization. The hybrid approach proved to be very effective when applied to a single objective optimization problems [17],[18]. This hybrid approach is an original concept whose elements and outline are presented in [16],[17],[18]. Application of this approach to multi-objective optimization has not been presented before. The best structure for the implementation of the above approach is a declarative CLP environment with operation research MP as a hybrid system. Furthermore, such a hybrid approach allows the use of all layers of the problem (data, structure, methods) to solve it. Finally, it allows the transformation of the problem (Section VIC) to such a form that can fully

exploit the strengths of the constraint propagation and data instances. It is well-known that there exist multiple non-dominated solutions for a multi-objective optimization problem. Those solutions are called “Pareto-optimal” solutions. In this paper, our objective is to obtain a “Pareto-optimal” set which provides evenly distributed Pareto solutions and it is convenient for the decision maker to select a suitable costs between production, distribution and environmental (F1 and F2) or between all costs and total distributor capacity (F1’ and F2’) etc. (Section VIA and Section VID). This hybrid programming framework is not just a blind attempt to integrate two environments, CLP/MP. The proposed approach is reinforced with the transformation, different representation of the problem (Section VIC) and using the algorithm for finding a “Pareto-optimal” set. In addition, hybridization refers to the class of decision problems which has certain property (This property is characterized by the constraints of many discrete decision variables and their summation).

V. THE CONCEPT AND IMPLEMENTATION ASPECTS OF THE HYBRID MULTI-OBJECTIVE PROGRAMMING FRAMEWORK

In this approach to the modeling and multi-objective optimization of supply chain problems, the hybrid multi-objective programming framework has been proposed, where:

- the decision and optimization models solved using the proposed framework can be formulated as a pure model of MOOP/MOLP or a hybrid model (with logical and non-linear constraints);
- knowledge related to the problem can be expressed as facts and constraints (linear, non-linear, logical and symbolic etc.);
- the problem is modeled in the constraint logic programming environment by CLP predicates, which is far more flexible than the MP environment;
- transforming the optimization model to explore its structure and data has been introduced by CLP predicates;
- optimization is performed by MP-based environments.
- the effective algorithm for finding a “Pareto-optimal” set has been introduced.

The schematic diagram of the implementation framework for the hybrid approach is presented in Figure 1. The names and descriptions of the predicates and procedures are shown in Table 1. From a variety of tools for the implementation of the CP-based environment, ECLⁱPS^e software [22] was selected. ECLⁱPS^e is an open-source software system for the cost-effective development and deployment of constraint programming applications. MP-based environment in implementation platform was LINGO by LINDO Systems [23]. LINGO Optimization Modeling Software is a powerful tool for building and solving mathematical optimization models. ECLⁱPS^e was used to implement the following predicates of the framework: CLP1, CLP2, CLP3 and CLP4 (Fig. 1, Table 1.). CLP predicates significantly restrict the space of feasible solutions (Fig. 1). The transformed files of

the model were transferred from ECLiPSe to LINGO where they were merged (MP1). Then the complete model was solved using LINGO efficient solvers (MP2). After that the model was modified for next step in procedure MP1 and solved again and again by MP2 in order to find a set of Pareto-optimal solutions (algorithm see appendix A).

TABLE I.

THE NAMES AND DESCRIPTIONS OF THE PREDICATES AND PROCEDURES

Symbol	Description
CLP1	The implementation of the model in CLP, the term representation of the problem in the form of predicates.
CLP2	Constraint propagation for the model or transformed 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.
CLP3	The transformation of the original problem aimed at extending the scope of constraint propagation. The transformation uses the structure of the problem and data. 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.
CLP4	Generation the model for mathematical programming-MOOP;
MP1	Merging files generated by CLP4 into one file. It is a model file format in LINGO system.
MP2	The solution of the model from the MP1 by MP solver. Sending the new solution to MP1 as a new constraint. Modification of the model. The process is repeated until the whole Pareto set is found (algorithm Appendix A).

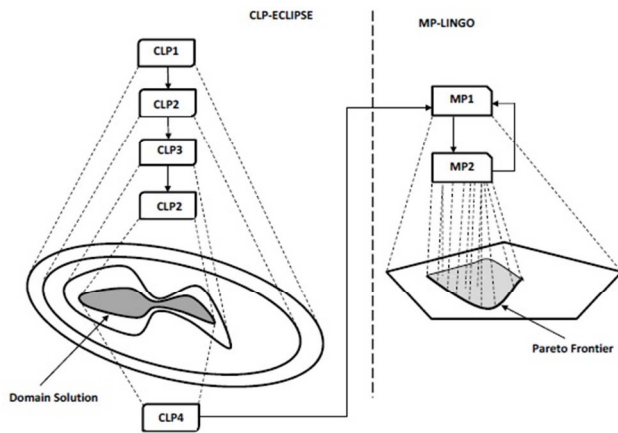


Fig. 1 The scheme of the hybrid multi-objective programming framework

VI. ILLUSTRATIVE EXAMPLES OF SUPPLY CHAIN MODELING AND MULTI-OBJECTIVE OPTIMIZATION - THE HYBRID APPROACH

The proposed approach was used and tested on two illustrative supply chain multi-objective optimization models. The model was formulated as a multi-objective mixed-integer optimization problem (MOOP) problem based on under constraints (2) .. (23) in order to test the proposed approach (Fig. 1) against the classical operation research approach. The models differ from one another objective

functions (Section 6.A). Indices, parameters and decision variables used in the models together with their descriptions are summarized in Table 2 and [16],[17]. The simplified structure of the supply chain network for this model, composed of producers, distributors and customers is presented in Figure 2. Both models are the cost multi-objective models that take into account 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 delivery and service by distributor, etc.), the transport mode and environmental aspects.

These models are based on optimization models taken from [16],[17].

TABLE II.

INDICES, PARAMETERS AND DECISION VARIABLES

Symbol	Description
<i>Indices</i>	
k	product type (k=1..K)
m	delivery point/customer/city (m=1..M)
n	manufacturer/factory (n=1..N)
e	distributor /distribution center (e=1..E)
d	mode of transport (d=1..D)
N	number of manufacturers/factories
M	number of delivery points/customers
E	number of distributors
K	number of product types
D	number of mode of transport
<i>Input parameters</i>	
F_e	the fixed cost of distributor/distribution center e
$C_{n,k}$	the cost of product k at factory n
$K1_{n,e,k,d}$	the variable cost of delivery of product k from manufacturer n to distributor e using mode of transport d
$A_{n,e,d}$	the fixed cost of delivery from manufacturer i to distributor s using mode of transport d
$K2_{e,m,k,d}$	the variable cost of delivery of product k from distributor e to customer m using mode of transport d
$G_{e,m,d}$	the fixed cost of delivery from distributor e to customer m using mode of transport d
Od_d	the environmental cost of using mode of transport d
<i>Decision variables</i>	
$X_{n,e,k,d,m}$	delivery quantity of product k from manufacturer n to distributor s using mode of transport d to customer m
$Xa_{n,e,k,d,m}$	if delivery is from manufacturer i to distributor e product k using mode of transport d to customer m then $Xa_{n,e,k,d,m} = 1$, otherwise $Xa_{n,e,k,d,m} = 0$
$Xb_{n,e,d}$	the number of courses from manufacturer n to distributor m using mode of transport d
$Y_{e,m,k,d}$	delivery quantity of product k from distributor e to customer m using mode of transport d
$Yb_{e,m,d}$	the number of courses from distributor e to customer m using mode of transport d
Tc_e	if distributor e participates in deliveries, then $Tc_e=1$, otherwise $Tc_e=0$

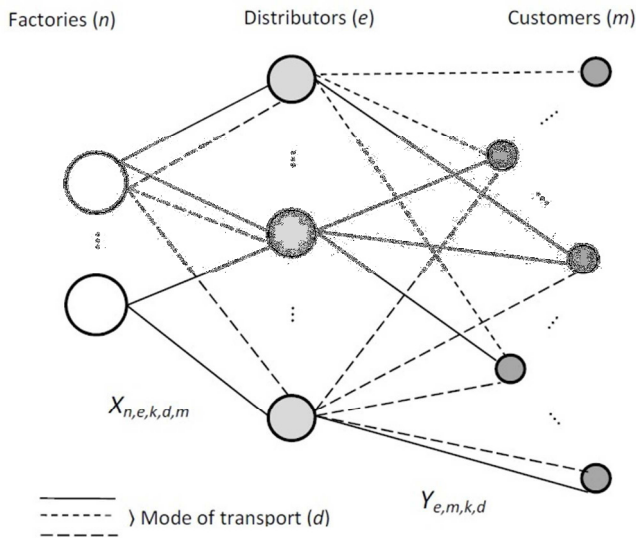


Fig. 2 Simplified supply chain network

A. Objective function

Two typical objective functions, F1 and F2, are commonly used in optimization issues. Objective function F1 (1a) defines the aggregate costs of the entire chain and consists of four 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 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 F1 determines the cost of manufacturing the product by the given manufacturer. The second objective function F2 (1b) 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 [8]. This hybrid approach and its implementation can be successfully used for other objective functions, including those named in Section 3. For the numerical examples from Section 7, in addition to the objective function formulated as above, the objective functions where $F1' = F1 + F2$ (sum (1a) and (1b)), whereas $F2' = V$ (total capacity of distribution centers) were formulated.

$$F1 = \sum_{e=1}^E F_e \cdot Tc_e + \sum_{n=1}^N \sum_{d=1}^D (A_{n,e,d} \cdot Xb_{n,e,d} + \sum_{k=1}^K (K1_{n,e,k,d} * X_{n,e,k,d,m})) + \quad (1a)$$

$$\sum_{e=1}^E \sum_{m=1}^M \sum_{d=1}^D (G_{e,m,d} \cdot Yb_{e,m,d} + \sum_{k=1}^K K2_{e,m,k,d} \cdot Y_{e,m,k,d}) + \sum_{n=1}^N \sum_{k=1}^K (C_{nk} \cdot \sum_{e=1}^E \sum_{d=1}^D \sum_{m=1}^M X_{n,e,k,d,m})$$

$$F2 = \sum_{d=1}^D Od_d (\sum_{n=1}^N \sum_{e=1}^E Xb_{n,e,d} + \sum_{e=1}^E \sum_{m=1}^M Yb_{e,m,d}) \quad (1b)$$

B. Two typical objective functions, F1 and F2, are commonly

The model was based on constraints (2) .. (23). Constraint (2) specifies that all deliveries of product k produced by the

manufacturer n and delivered to all distributors e using mode of transport d do not exceed the manufacturer's production capacity. Constraint (3) covers all customer m demands for product k through the implementation of delivery by distributors s (the values of decision variables $Y_{e,m,k,d}$). The flow balance of each distributor e corresponds to constraint (4). The possibility of delivery is dependent on the distributor's technical capabilities (5). Time constraint (6) ensures the terms of delivery are met. Constraints (7), (8), (9), (10), (11) guarantee deliveries with available transport taken into account. Constraints (12), (13), (14) set values of decision variables based on binary variables Tc_e , $Xa_{n,s,k,d,m}$, $Ya_{e,m,k,d}$, $X_{n,e,k,d,m}$, $Y_{e,m,k,d}$. Dependencies (15) and (16) represent the relationship based on which total costs are calculated. In general, these may be any linear functions. The remaining constraints (17) .. (23) arise from the nature of the model (MILP). A detailed description of the constraints and their formalization have been presented in [16],[17].

C. Model transformation

The ability to transform the problem using CLP is one of the most important features of the hybrid programming framework. Due to the nature of the decision problem (adding up variables in the objective function and constraints), 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 programming framework (Fig. 1). 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 CLP predicates with parameters. While modeling problem (1) .. (23), quantities n , e , k , $d1$, $d2$, m and decision variables $X_{n,e,k,d1,m}$ and $Y_{e,m,k,d2}$ were predicate parameters (Fig. 3a). The process of finding the solution may consist in using the constraints propagation methods, labeling of variables and the backtracking mechanism [15]. The quality of constraints propagation and the number of backtrackings are affected to a high extent by the number of parameters that must be specified/labeled in the given predicate. In the models presented above, the classical problem representation included four parameters (Fig. 3b): n , e , $d1$, $d2$, and two decision variables $X_{n,e,k,d1,m,j}$, $Y_{e,m,k,d2}$. Considering the domain size of each parameter, the process is complex and time-consuming. The idea was to transform the problem by changing its representation without changing the very problem. All permissible routes were first generated by CLP3 predicate based on the fixed data (factories, distributors, mode of transport etc.) and a set of orders (Fig. 4a), then the specific values of parameters n , e , k , $d1$, $d2$, m were assigned to each of the routes. Thereby only one parameter of the transformed decision variable $X_{n,k,e,j,d1,d2}^T$ (deliveries) had to be specified (Fig. 4b). This transformation fundamentally improved the efficiency of the

constraint propagation and reduced the number of backtracks.

Solution (Objective_Function, parameters) :- parameters /O_n,P,M,D,F,Tu,Tu,Oq,X,Y,T/

Fig. 3a. Representation of the problem in the classical approach-the main search predicate

Solution ($V_{FC}^1, o_1, p_1, m_1, \dots, 12, \dots, 8$)
 ($V_{FC}^1, o_2, p_2, m_2, \dots, 20, \dots, 6$)
 ($V_{FC}^1, o_4, p_3, m_2, \dots, 12, \dots, 9$)
 ...

Fig. 3b. Representation of the problem in the classical approach- the process of finding a solution

Solution_hybrid (Objective_Function, parameters) :- parameters /route_n,P,M,D,F,Tu,Tu,Oq,X^T, T/

Fig. 4a. Representation of the problem in the hybrid approach- the main search predicate

Solution_hybrid ($V_{FC}, ((route_1, f_1, p_1, c_1, m_1, s_1, s_1, 5, 10, 100, 8)$
 ($route_2, f_1, p_2, c_1, m_1, s_1, s_1, 5, 15, 120, 12$),
 ($route_2, f_1, p_2, c_1, m_1, s_2, s_1, 5, 15, 80, 8$),
 ($route_4, f_2, p_2, c_2, m_1, s_1, s_2, 5, 12, 20, 6$), ...), ...)

Fig. 4b. Representation of the problem in the hybrid approach- set of feasible routes

Symbols used in descriptions are presented in Table III.

TABLE III.

INDICES, SYMBOLS USED IN THE REPRESENTATION OF THE PROBLEM

Symbol	Description
V_{FC}	Value of the objective function calculated on the basis of the vector of parameters.
O_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, $Tu \in \{s_1, s_2, \dots, s_i\}$.
T	Delivery time/period.
Oq	Order quantity.
$X/Y/X^T$	Delivery quantity.
route_n	Routes name-number.
,	Separates predicate parameters.
:-	Separates predicate heading from its definition.
_	Unknown value of the variable.

The obtained multi-objective optimization model after the transformation (MOOPT) has different decision variables and different constraints than those in the MOOP (1) .. (23). Some of the decision variables are redundant; other variables are subject to aggregation. This results in a very large reduction in their number. Decision variables before and after the transformation are shown in Table IV. The transformation also reduces or eliminates some of the constraints of the model. Thus constraints (4), (6), (12), (13), (14), (15) and (16) present in the MOOP (1) .. (23) are redundant in the MOOPT. Balance constraint (4) is unnecessary because the route defines the specific distribution center. Only those routes are generated that meet the time constraints, therefore constraint (6) does not make

sense. Binariness ensures whether or not the route occurs, thus constraint (12) is redundant. Reduction of certain variables also affects the reduction of constraints, hence lack of constraints (13), (14) in the model. Constraints (15) and (16) are unnecessary, because the delivery costs are now calculated for the entire route.

TABLE IV.
DECISION VARIABLES USED IN THE MOOP AND TRANSFORMED MOOPT MODELS

MOOP	MOOPT	Description of the decision variables after the transformation
$X_{n,e,k,d_1,m}$	X_{n,k,e,m,d_1,d_2}^T	Decision variable X^T , unlike the initial decision variables X, Y , is generated only for technologically possible indices combinations. It defines the allocation size of product k to the route of deliveries.
Y_{e,m,k,d_2}		After transformation replaced by the appropriate factor for the route - generated by the CLP.
$Xa_{n,e,k,d,m}$	unnecessary	Without change, the same sense.
$Xb_{n,e,d}$	$Xb_{n,e,d}$	Without change, the same sense.
$Ya_{e,n,k,d}$	unnecessary	After transformation replaced by the appropriate factor for the route - generated by the CLP.
$Yb_{e,m,d}$	$Yb_{e,m,d}$	Without change, the same sense.
Tc_e	Tc_e	Without change, the same sense.

After the transformation in the MOOPT model, the objective functions $F1$ and $F2$ were re-formulated. New objective functions, $F1T$ (A1a) and $F2T$ (A1b) were obviously formulated using new decision variables (Table IV) and calculated parameters by CLP (Table V). These parameters were determined as a result of constraint propagation and the transformation itself using CLP2 and CLP3. Owing to these quantities, it is possible to introduce to the MOOPT model additional constraints (A2) ... (A7). These constraints affect the efficiency of the search for a solution by narrowing down the search area. Table VI describes these constraints.

$$FIT = \sum_{e=1}^E Tc_e \cdot F_e + \sum_{n=1}^N \sum_{e=1}^E \sum_{d=1}^D (Xb_{n,e,d} \cdot Ksc_{n,e,d}) + \sum_{e=1}^E \sum_{m=1}^M \sum_{d=1}^D (Yb_{e,m,d} \cdot Ksm_{e,m,d}) + \sum_{n=1}^N \sum_{k=1}^K \sum_{e=1}^E \sum_{m=1}^M \sum_{d_1=1}^D \sum_{d_2=1}^D (X_{n,k,e,m,d_1,d_2} \cdot Kz_{n,k,e,m,d_1,d_2}) \quad (A1a)$$

$$F2T = \sum_{d=1}^D Od_d \cdot \left(\sum_{n=1}^N \sum_{e=1}^E Xb_{n,e,d} + \sum_{e=1}^E \sum_{m=1}^M Yb_{e,m,d} \right) \quad (A1b)$$

$$\sum_{n=1}^N \sum_{e=1}^E Xb_{n,e,d} + \sum_{e=1}^E \sum_{m=1}^M Yb_{e,m,d} \geq R \min_d \text{ for } d = 1..D \quad (A2)$$

$$\sum_{n=1}^N \sum_{e=1}^E Xb_{i,s,d} + \sum_{e=1}^E \sum_{m=1}^M Yb_{e,m,d} \leq R \max_d \text{ for } d = 1..D \quad (A3)$$

$$\sum_{n=1}^N \sum_{e=1}^E \sum_{d=1}^D Xb_{n,e,d} \geq \text{Min}_{F_C} \quad (A4)$$

$$\sum_{e=1}^E \sum_{j=1}^M \sum_{d=1}^D Yb_{e,m,d} \geq \text{Min}_{D_C} \quad (A5)$$

$$\sum_{n=1}^N \sum_{e=1}^E \sum_{d=1}^D Xb_{n,e,d} + \sum_{n=1}^N \sum_{m=1}^M \sum_{d=1}^D Yb_{e,m,d} \geq \text{Min}_{TU} \quad (2)$$

$$\sum_{e=1}^E Tc_e \geq Cn \quad (A7)$$

TABLE V.
CALCULATED FIGURES FOR MOOPT

Symbol	Description
Rmin _d	Minimum number of transport units d (CLP – propagation).
Rmax _d	Maximum number of transport units d (CLP – propagation).
Min_F_C	Minimum number of transport units in the route Factories – Centers (CLP – propagation).
Min_D_C	Minimum number of transport units in the route Distributors – Customers (CLP – propagation).
Min_TU	Minimum number of transport units (CLP – propagation).
Cn	Minimum number of active centers (CLP – propagation).
Ksc	The fixed cost of delivery from manufacturer to distributor by using transport mode (CLP-calculated based on fixed data).
Ksm	The fixed cost of delivery from distributor to customer by using transport mode (CLP-calculated based on fixed data).
Kz	The variable cost of delivery (CLP-calculated based on fixed data).

TABLE VI.
ADDITIONAL CONSTRAINTS FOR MOOPT

Constraints	Description
A2	narrowing the size of the transport unit domain from the bottom
A3	narrowing the size of the transport unit domain from the top
A4	the minimum number of all transport unit types necessary for the shipment from the factory to the distribution center
A5	the minimum number of all transport unit types necessary for the shipment from the from the center to customers
A6	the minimum number of transport units in routes
A7	the number of working distribution centers

D. Decision support

Implementation of the presented models using implementation hybrid programming framework can support decision-making in the following practical areas of supply chain (not limited to the following):

- the multi-optimization of total cost of the supply chain (objective functions F1 and F2/ F1' and F2' -Table VII) in the form of the Pareto-optimal solution set (Fig.5.);
- analysis of delivery costs with environmental cost optimization;
- the selection of the transport fleet number, capacity and modes for specific total costs (Fig.6, Fig.8);
- the sizing of distributor warehouses and the study of their impact on the overall costs (objective functions F1' and F2' -Table VII and Fig.7);
- the sizing production capacity and the study of their impact on the overall costs;
- the selection of transport routes for elements of the Pareto-optimal solution set;

VII. NUMERICAL EXPERIMENTS AND ANALYSIS

In order to verify and evaluate the proposed hybrid approach and implementation platform, many numerical

experiments were performed. All the examples relate to the supply chain with seven manufacturers ($n=1..7$), three distributors ($e=1..3$), ten customers ($m=1..10$), three modes of transport ($d=1..3$), and twenty types of products ($k=1..20$). The numerical data were taken from the trans-regional distributor FMCG and transportation fleet parameters available online. Experiments began with four examples of P1 .. P4 for the optimization MOOP model (1) .. (23). The examples differ the number of orders (No). The first series of experiments was designed to show the benefits and advantages of the presented approach. For this purpose the model was implemented in both the hybrid programming framework (MOOPT) and mathematical programming environment (MOOP). In the next stage of the experiments, the objective function was changed (Section 6.A). Its first element F1' refers to the total cost of deliveries whereas F2' defines the distributor's total storage capacity available.

Due to the nature of the optimization problems considered here, a new algorithm based on the ϵ -constraint method was proposed in the final phases of MP1 and MP2. This algorithm (Appendix A) helped determine a set of Pareto-optimal solutions. The algorithm has been implemented in LINGO by using meta-modeling and programming language LINGO package.

The detailed results for a 20-order example (P4) are shown in Table 7. Figures 5 and 6 show the corresponding sets of Pareto-optimal solutions for a 20-order example. It is evident that only the hybrid approach provides the results within the acceptable time (Table VIIIA and VIIIB).

This is possible owing to the reduction of the problem size and, in particular, transformation of the problem with the use of methods from both environments, MP and CLP. For the illustrative examples discussed here, a 23-fold reduction in the number of constraints (C) was obtained with an over 130-fold reduction in the number of decision variables (V). This gives the size of the combinatorial problem calculated as $V \times C$ reduced by more than 2600 times. Comparison of the results (Table 8a and 8b) from the hybrid approach with those from mathematical programming indicates that, first, when the same model and data are used in the classical manner (mathematical programming), the Pareto-optimal solution was obtained only for the smallest example P1 (5 orders). Second, for larger examples P2, P3, P4, it was hardly possible to find at least one point of the Pareto-optimal solution in acceptable time (computing stopped after 600 s). Comparing the proposed approach to modeling and solution in only CLP environment is pointless due to the nature and weak capacity of the CLP relative to the optimization of problems where many variables are added up, which is illustrated in [16].

TABLE VII.
THE DETAILED RESULT FOR EXAMPLE P4 (No=20)

PP	F1	F2	PP	F1'	F2'=V
1	F1 \geq 26540	F2=3775	1	F1'=0	F2' \leq 0
2	F1 \geq 29345	F2=3725	2	F1'=0	F2' \leq 500
3	F1 \geq 32150	F2=3725	3	F1'=0	F2' \leq 1000
4	F1 \geq 34955	F2=3655	4	F1'=0	F2' \leq 1500
5	F1 \geq 37760	F2=3655	5	F1'=45195	F2' \leq 2000
6	F1 \geq 40565	F2=3655	6	F1'=35705	F2' \leq 2500

7	$F1 \geq 43370$	$F2 = 3655$	8	$F1' = 35705$	$F2' \leq 3000$
8	$F1 = 33670$	$F2 \geq 3655$	9	$F1' = 35705$	$F2' \leq 3500$
9	$F1 = 26540$	$F2 \geq 4158$	10	$F1' = 35705$	$F2' \leq 4000$
10	$F1 = 26540$	$F2 \geq 4661$	PP-the number of element of Pareto-optimal set $F1, F2, F1', F2'$ -objective function		
11	$F1 = 26540$	$F2 \geq 5164$			
12	$F1 = 26540$	$F2 \geq 5668$			
13	$F1 = 26540$	$F2 \geq 6171$			
14	$F1 = 26540$	$F2 \geq 6075$			

TABLE VIIIA.

THE PARAMETERS OF THE PROCESS OF FINDING A SET OF PARETO-OPTIMAL SOLUTIONS FOR ILLUSTRATIVE EXAMPLES (OBJECTIVE FUNCTIONS $F1$ AND $F2$)

Example	No	MP-based approach (MOOP)			Hybrid programming framework (MOOPT)		
		T	V	C	T	V	C
P1	5	53	30369	16177	7	223	693
P2	10	----	30369	19122	10	264	697
P3	15	----	30369	20067	14	287	702
P4	20	----	30369	21012	44	318	703

* calculations stopped after 600 s, not found even one point from the set of Pareto-optimal solutions
T-solution time, V- the number of integer variables, C- the number of constraints

TABLE VIIIB.

THE PARAMETERS OF THE PROCESS OF FINDING A SET OF PARETO-OPTIMAL SOLUTIONS FOR ILLUSTRATIVE EXAMPLES (OBJECTIVE FUNCTIONS $F1'$ AND $F2'$)

Example	No	MP-based approach (MOOP)			Hybrid programming framework (MOOPT)		
		T	V	C	T	V	C
P1	5	70	30369	16179	11	223	691
P2	10	----	30369	19124	14	264	695
P3	15	----	30369	20069	21	287	700
P4	20	----	30369	21014	36	318	701
P5	25	----	30369	21959	67	357	707

* calculations stopped after 600 s, not found even one point from the set of Pareto-optimal solutions
T-solution time, V- the number of integer variables, C- the number of constraints

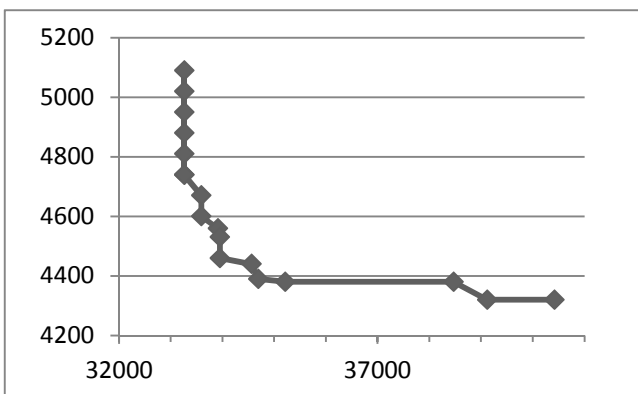


Fig. 5 A set of Pareto-optimal solutions for illustrative example P4 (No=20, vertical axis $F2$ horizontal axis $F1$)

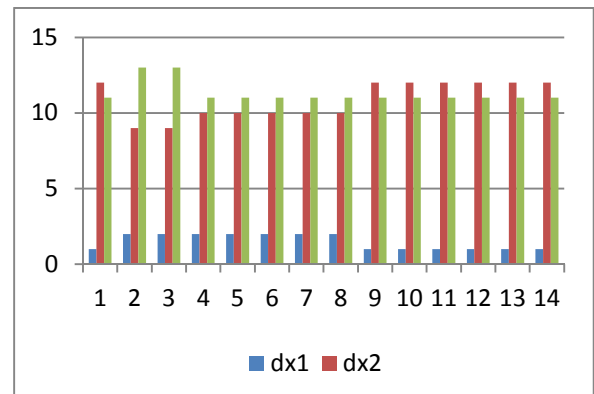


Fig. 6 The use of mode of transport for illustrative example P4 (No=20, $F1, F2$, vertical axis the number of mode of means of transport-dx1,dx2,dx3, horizontal axis Pareto set of points -PP)

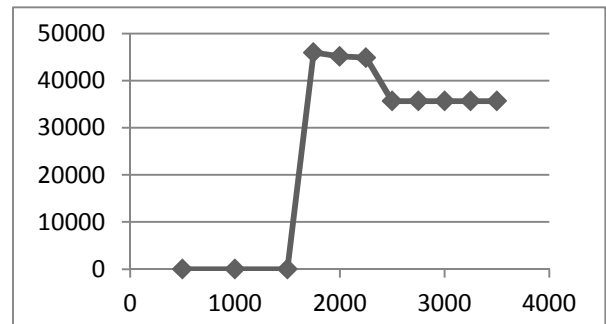


Fig. 7 A set of Pareto-optimal solutions for illustrative example E4 (No=20, vertical axis $F2'$ horizontal axis $F1'$)

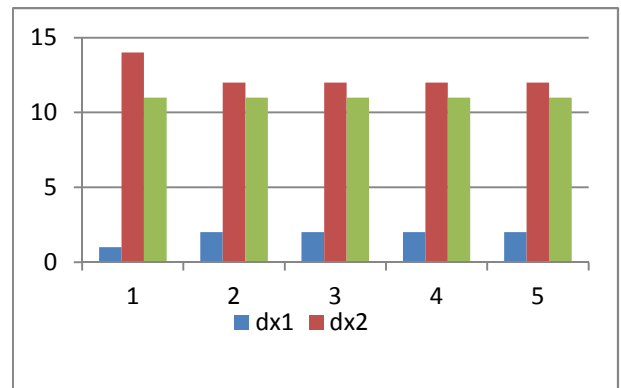


Fig. 8 The use of mode of transport for illustrative example P4 (No=20, $F1', F2'$, vertical axis the number of mode of means of transport-dx1,dx2,dx3, horizontal axis Pareto set of points -PP for $F1' > 0$)

VIII. CONCLUSIONS

This hybrid programming framework is especially significant and suited for multi-objective optimization, where a slightly changed single-objective problem has to be solved multiple times and where the modeling efficiency and ease are essential. The efficiency of the proposed approach is based on the reduction of the combinatorial problem. This means that using the hybrid approach practically for all models of this or a similar class, the same or better solutions are found even up to two hundred times faster (the optimal

instead of the feasible solutions). Another element contributing to the high efficiency of the method is a possibility to determine the values or ranges of values for some of the decision variables (predicate CLP2). The presented transformation of the problem (predicate CLP3), characteristic of the problems that have the structure as in Fig. 2, is an important aspect of this approach. It should be emphasized that with this approach it is possible not only to solve optimization problems faster, but also to solve much larger problems than in the [24]. The proposed solution is highly recommended for all types of decision problems in supply chain or for other problems with a similar structure. This structure is characterized by the constraints of many discrete decision variables and their summation. Furthermore, this method can model and solve problems with logical constraints. Therefore the implementations in the form of hybrid platform can be applied to various practical decision problems in the area of logistics, transport, production, scheduling or project management. In addition to the undoubted effectiveness of the proposed declarative hybrid approach, we should underline the possibility of modeling decision problems.

Further work will focus on running the optimization models with non-linear and other logical constraints, uncertainty, fuzzy logic [25] etc., numerical test with hybrid models (HM) and different scheduling problems and resource allocation and activity coordination in the supply chain [26]. It is also planned to implement the framework in the form of cloud applications [28].

APPENDIX ALGORITHM FOR FINDING A SET OF PARETO-OPTIMAL SOLUTIONS.

```

Enter the step (for how many points divided
interval)
Number_of_steps =? /input value/
Solve (min objective F1
  subject to
    the constraints of 1 - 27 (primary problem)
    or A1-A7 (problem after transformation)
)
Save the F1min-determined value of the objective
function
F2max = F2
Solve (min objective F2
  subject to
    the constraints of 1 - 27 (primary problem)
    or A1-A7 (problem after transformation)
)
Save the F2min-determined value of the objective
function
F1max = F1
There are designated intervals of a function F1
and F2
(F1min, F1max) and (F2min, F2max)
Optimization of F2
discretization = (F1max - F1min) / Number_of
_steps
cutting_off = F1min
i = 1
WHILE (cutting_off < F1max)
{
  Solve (min F2
    subject to

```

```

the constraints of 1 - 27 (primary problem)
or A1-A7 (problem after transformation)
F1> = cutting_off
)
save the pareto-optimal point
F2 (i) - optimal value of the objective function
F1(i) = F1
cutting_off cutting_off + = discretization
i = i + 1
}
Optimization of F1
discretization = (F2max - F2min) / Number_of
_steps
cutting_off = F2min
WHILE (cutting_off < F2max)
{
  Solve (min F1
    subject to
      the constraints of 1 - 27 (primary problem)
      or A1-A7 (problem after transformation)
    F2> = cutting_off
  )
  save the point
  F1 (i) - optimal value of the objective function
  F2 (i) = F2
  Cutting_off=cutting off + discretization
  i = i + 1
}
F1=F1 or F1'
F2=F2 or F2'

```

REFERENCES

- [1] Beamon B.M, Supply chain design and analysis: models and methods, International Journal of Production Economics 55, 281–294, 1998.
- [2] Mula J., Peidro D., Diaz-Madronero M., Vicens E., Mathematical programming models for supply chain production and transport planning, European Journal of Operational Research, 204, 377–390, 2010.
- [3] Gunasekaran A., Ngai EWT., Modeling and analysis of build-to-order supply chains. Eur J Oper Res 2009;195(2):319–34, 2009.
- [4] Howard M, Miemczyk B.J, Graves A., Automotive supplier parks: an imperative for build-to-order. J Purch Supply Manage 2006, 12, 91–104, 2006
- [5] Krajewski L., Wei J.C., Tang L.L., Responding to schedule changes in build-to-order supply chains. J Oper Manage 2005;23(5):452–69, 2005.
- [6] Abdolhossein S., Napsiah I., Norzima Z., Ariffin M. K. A., Nezamabadi-pour H., Mirabi H., A Multiobjective Optimization Model in Automotive Supply Chain Networks, Mathematical Problems in Engineering, vol. 2013, Article ID 823876, doi:10.1155/2013/823876, 2013.
- [7] Minor P., Hertwin O., Elías O.B., Ruben T.O., Luis M., Variations in the Flow Approach to CFCLP-TC for Multiobjective Supply Chain Design, Mathematical Problems in Engineering, vol. 2014, Article ID 816286, doi:10.1155/2014/816286, 2014.
- [8] Seyed M, Al-e-Hashem J.M., Aryanezhad M.B, Sadjadi S.J., An efficient algorithm to solve a multi-objective robust aggregate production planning in an uncertain environment, The International Journal of Advanced Manufacturing Technology, January 2012, Volume 58, Issue 5-8, 765-782, 2012.
- [9] Shankar L., Basavarajappa B., Chen S., Jason C.H, Kadadevaramath, Rajeshwar S., Location and allocation decisions for multi-echelon supply chain network - A multi-objective evolutionary approach, Expert Systems with Applications, Volume 40, Issue 2, 1 February 2013, 551–562, 2013
- [10] Che Z.H., Chiang C.J., A modified Pareto genetic algorithm for multi-objective build-to-order supply chain planning with product assembly, Advances in Engineering Software 41, 1011–1022, 2010.
- [11] Wang F., Lai X., Shi N., A multi-objective optimization for green supply chain network design, Decision Support Systems, 51, 262–269, 2011.

- [12] Erenguc S. S., Simpson N. C., Vakharia A. J., Integrated production/distribution planning in supply chains: an invited review, *European Journal of Operational Research*, 115, 219–236, 1999.
- [13] Collette Y., Siarry P., *Multiobjective Optimization, Principles and Case Studies*, Springer, IX, 293 p, 2003.
- [14] Apt K., Wallace M., *Constraint Logic Programming using Eclipse*, Cambridge University Press, 2006
- [15] Rossi F., Van Beek P., Walsh T., *Handbook of Constraint Programming (Foundations of Artificial Intelligence)*, Elsevier Science Inc. New York, NY, USA, 2006
- [16] Sitek, P., Wikarek, J., A hybrid approach to modeling and optimization for supply chain management with multimodal transport, *IEEE Conference: 18th International Conference on Methods and Models in Automation and Robotics (MMAR)*, 777-782, 2013.
- [17] Sitek, P., A hybrid CP/MP approach to supply chain modelling, optimization and analysis, *Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2014, pp.1345-1352. DOI: 10.15439/2014F89.
- [18] Sitek, P., Wikarek, J.: A Hybrid Approach to the Optimization of Multiechelon Systems. *Mathematical Problems in Engineering* vol. 2015, Article ID 925675, 12, pages, 2015. doi:10.1155/2015/925675.
- [19] Milano M., Wallace M., Integrating Operations Research in Constraint Programming, *Annals of Operations Research*, vol. 175 issue 1, 37-76, 2010.
- [20] 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, 6-20, 2008.
- [21] Bockmayr A., Kasper T., Branch-and-Infer, A Framework for Combining CP and IP, *Constraint and Integer Programming Operations Research/Computer Science Interfaces Series*, Volume 27, 59-87, 2004.
- [22] Eclipse - The Eclipse Foundation open source community website, www.eclipse.org, 2015
- [23] LINDO Systems - Optimization Software: Integer Programming, Linear Programming, Nonlinear, www.lindo.com, 2015
- [24] Sitek P., Wikarek J., Cost optimization of supply chain with multimodal transport, *Federated Conference on Computer Science and Information Systems (FedCSIS)*, 1111-1118, 2012.
- [25] M. Relich and W. Muszynski, "The use of intelligent systems for planning and scheduling of product development projects", *Procedia Computer Science*, vol. 35, pp. 1586–1595, 2014.
- [26] Grzybowska K., Selected Activity Coordination Mechanisms in Complex Systems, J. Bajo et al. (Eds.): *PAAMS 2015 Workshops, CCIS 524*, Springer International Publishing Switzerland, pp. 1–11, 2015.
- [27] Grzybowska K., Kovács G., Logistics Process Modelling in Supply Chain – algorithm of coordination in the supply chain – contracting, *International Joint Conference SOCO'14-CISIS'14-ICEUTE'14, Advances in Intelligent Systems and Computing*, Vol. 299, pp. 311-320, 2014
- [28] Bağ, S., Czarnecki R., Deniziak S. (2013). Synthesis of Real-Time Cloud Applications for Internet of Things. *Turkish Journal of Electrical Engineering & Computer Sciences*, DOI: 10.3906/elk-1302-178.
- [29] Bocewicz, G., Nielsen, I., Banaszak, Z. (2014). Iterative multimodal processes scheduling. *Annual Reviews in Control* 38(1), 113-132.