




# A multi-level approach to ubiquitous modeling and solving constraints in combinatorial optimization problems in production and distribution

Paweł Sitek<sup>1</sup>  · Jarosław Wikarek<sup>1</sup>

Published online: 29 December 2017  
© The Author(s) 2017. This article is an open access publication

## Abstract

Constraints, although ubiquitous in production and distribution planning, scheduling and control, often lead to inconsistencies in the decision-making process. The constraint-based modeling helps circumvent many organization-impacting issues. To address this, we developed a multi-level approach to the modeling and solving of combinatorial optimization problems. It is versatile and effective owing to the use of multi-level presolving and multiple paradigms, such as constraint programming, logic programming, mathematical programming and fuzzy logic, for their complementary strengths. The capability of this framework and its advantage over mathematical programming alone or over hybrid frameworks is shown in the illustrative example, in which combinatorial optimization is used as a benchmark to prove the effectiveness of the proposed approach. Knowledge of the problem is stored in the form of facts.

**Keywords** Constraint logic programming · Mathematical programming · Constraint satisfaction problem ubiquitous modeling and solving constraints · Presolving · Hybrid methods · Manufacturing and distribution

## 1 Introduction

Constraints are ubiquitous in various areas of production and distribution. They may correspond to materials, technologies, resources, time, capacity, interoperation transport, etc. (production) or to available storage capacity, selection of distributors, transport duration, number of transport means, their types, etc. (distribution). Proper constraint modeling and solving will have a considerable impact on production-distribution planning and control and will allow decision-makers to detect difficult situations early enough to handle them safely. If the constraints are modeled and solved, fully or partially, at the moment they appear in a given area or for a given problem, but not collectively in subsequent stages, the global decision or optimization model will be simpler and solving it will take less time. We can thus talk about ubiquitous modeling and solving of constraints (UMSC) in these problems.

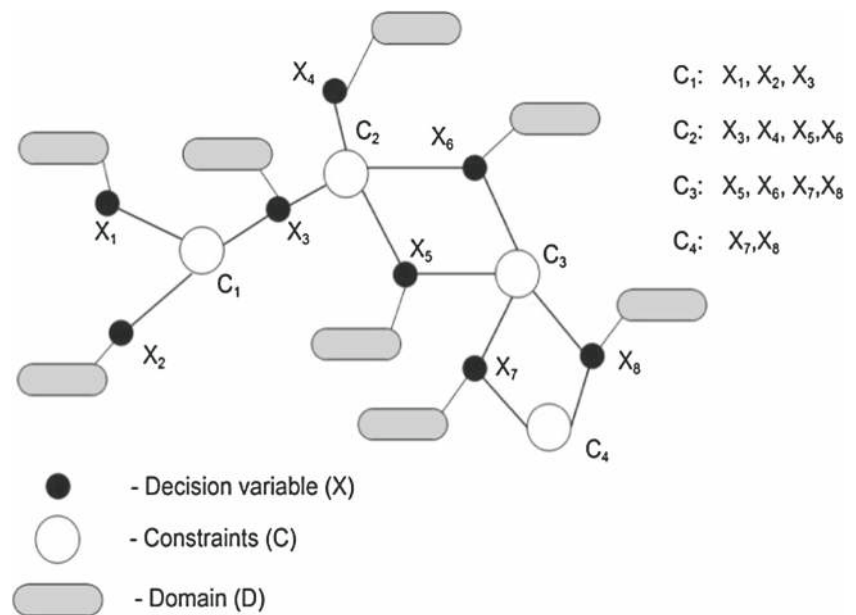
The concept of UMSC can be the basis for creating decision support systems and process optimization and control at various levels. Planning, scheduling and resource allocation problems are usually modeled using OR (operation research) approaches such as mathematical programming (MP) methods, with linear programming (LP) used at the strategic or tactical level (aggregated level). More detailed planning, scheduling and resource allocation require integer and binary decision variables and to this end, MIP (Mixed Integer Programming), IP (Integer Programming) or MILP (Mixed Integer Linear Programming) Schrijver [17] models have to be employed. The resulting models are very complex. To solve them, substantial computing effort is necessary. Real problems may become NP-hard problems. In the 1990s, constraint programming-based environments appeared Rossi et al. [16]; Benhamou et al. [3]; Tsang [25]; Apt [1] and their implementations to production and distribution problems were reported Liess and Michelon [12]; Rocha and Ramos [15]; Bocewicz and Banaszak [4]. Enormous flexibility, easiness and the range of constraint modeling gave them definite advantage over OR-based environments. But only at the modeling stage – CP-based environments turned out to be less effective at the solving stage, in particular, in solving combinatorial optimization problems. Their effectiveness often depends on the structure of modeled constraints and data, or on decision variable

---

✉ Paweł Sitek  
sitek@tu.kielce.pl  
Jarosław Wikarek  
j.wikarek@tukielce.pl

<sup>1</sup> Department of Control and Management Systems, Kielce University of Technology, Kielce, Poland

**Fig. 1** Constraint network (CN) in CSP



domain range. This issue does not exist in the MP-based environments, where model-solving methods are independent of both constraint structure and data. As for the UMSC concept, the approach that combines multiple paradigms such as CP- constraint programming, LP- logic programming, MP-mathematical programming, FL-fuzzy logic, etc. is the best solution.

The paper is organized as follows. Section 2 describes the constraint-based methods. The multi-level approach to ubiquitous constraints modeling and solving is presented in Section 3. The illustrative example and computational results of the proposed approach are described in Sections 4 and 5. Final remarks and possible future development and applications are shown in Section 6.

## 2 Constraint-driven programming, mathematical programming and integrated methods – basic principles

For the ubiquitous modeling and solving of different types of constraints, the most effective and most flexible are constraint-driven methods and environments based mostly on constraint satisfaction problems (CSPs) Tsang [25]; Apt [1]. A constraint satisfaction problem (CSP) is defined by a set of decision variables,  $X_1, X_2, \dots, X_n$ , and a set of constraints,  $C_1, C_2, \dots, C_m$ . Each decision variable  $X_i$  has a nonempty domain  $D^{X_i}$  of possible values. Each constraint  $C_i$  involves some subset of the decision variables and specifies the feasible combinations of values for that subset. A state of the problem is defined by an assignment of values to some or all of the decision variables,  $\{X_i = v_i, X_j = v_j, \dots\}$ . An assignment that does not violate any constraints

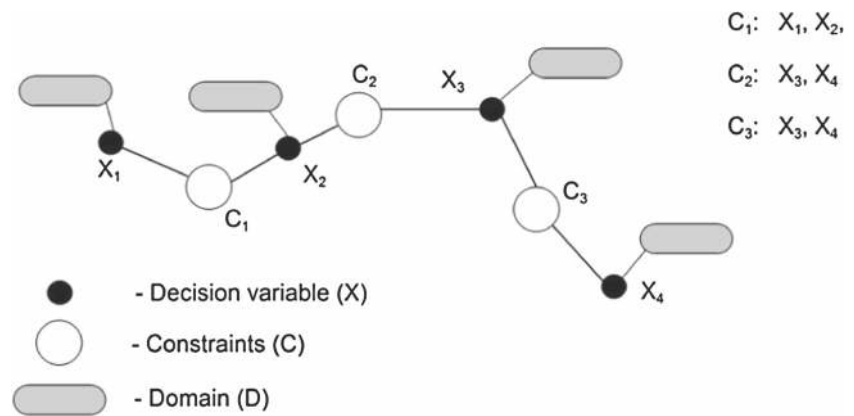
is called a consistent or legal assignment. A complete assignment is one in which every variable is mentioned, and a solution to a CSP is a complete assignment that satisfies all the constraints. Some CSPs also require a solution that maximizes an objective function than we speak about COP (Constraint Optimization Problem).

The CSP is typically determined on a constraint network (Fig. 1). A constraint network (CN) comprises a set of decision variables, each associated with a domain of values and a set of constraints. A CN defines the feasible combination of values of decision variables through a constraint subset. Formally, constraints are functions that effectively define the assignment of decision variables to some domain (Figs. 1 and 2).

Several modifications to the classic CSP model have been proposed: Flexible CSPs (relaxation of assumption that each solution must satisfy all constraints, including Fuzzy\_CSP and Weighted\_CSP), Dynamic\_CSPs (a sequence of static CSPs) and Decentralized\_CSPs. CSPs are to solve problems by modeling constraints, which represent all properties and requirements of the problem, and by finding solutions that satisfy all these constraints. CSP on finite domains are usually solved using different forms of search such as local search, constraint propagation and many variants of backtracking methods Rossi et al. [16]. A general search algorithm for solving a CSP is shown in Fig. 3. The highest effectiveness of CSPs is reported for binary constraint problems, where each constraint binds only two decision variables (Fig. 2).

For constraints that bind more than two decision variables, constraint propagation effectiveness decreases significantly and the number of backtrackings increases rapidly Sitek and Wikarek [20]. This feature makes the CSP-based

**Fig. 2** Constraint network in CSP for binary constraints



environment, such as CP (Constraint Programming) and CLP (Constraint Logic Programming) less effective in solving complex problems, such as planning, scheduling or resource allocation. The structure of the constraints has no effect on the effectiveness of the MP methods, in contrast to a large number of integer decision variables, which, when present, reduce it substantially. Both MP and CLP involve decision variables and constraints. However, the types of the decision variables and constraints that are used and the way the constraints are solved are different in the two approaches Bockmayr and Kasper [5]; Hooker [11]; Barth and Bockmayr [2]. MP methods take into account only linear constraints (equations and inequalities) which include binary, integer and continuous decision variables. In the constraint-driven approach, the programming language is richer in terms of the types of constraints. In addition to linear equations and inequalities, there are various other constraints: nonlinear, disequalities, and symbolic (disjunctive, exclude, cumulative, alldifferent, profile etc.) Rossi et al. [16]. Moreover, CLP is less effective in combinatorial optimization problems Escudero et al. [9] that often occur in planning, scheduling, and control in manufacturing and distribution.

To sum up, in mathematical programming, the sets of constraints (equations) describe the problem but do not indicate how to solve it. In constraint programming, each constraint invokes a procedure that screens out unacceptable solutions. The most important elements that decide the effectiveness of mathematical programming include: relaxation, tools for filtering and duality theory Schrijver [17]. Relaxation methods in MP tend to be more effective when constraints and/or objective functions

contain many decision variables. In general, MP relies on numerical calculation, which increases its effectiveness. In contrast, constraint-based environments (CP/CLP) may fail when constraints contain many decision variables. This follows from the fact that these environments (CP/CLP) are based on logic processing and the constraints do not propagate well. Moreover, CP/CLP are often insufficient for finding optimal solutions due to lack of relaxation technology and numerical calculation. However, the CP/CLP environments are a more powerful modeling language with which any type of constraints can be easily modeled and they use the structure within a problem (horizontal structure), thus contrary to MP with real vertical structure (a model is independent from data). Also, the CP/CLP environments have built in methods for removing infeasible values from variable domains (filtering, domain consistency, constraint propagation, etc). It is clear that both approaches (MP and CP/CLP) are complementary in many aspects and areas.

With this in mind, research has been undertaken to attempt integrating these paradigms. Several scenarios of their integration were reported in the literature Hooker [11]:

- *Double modeling* - use both CLP and MP models and exchange information while solving.
- *Search-inference duality* - view CLP and MP methods as special cases of a search/inference duality.
- *Decomposition* - decompose problems into a CLP part and an MP part using a Benders scheme (Benders decomposition integrates two solution methods: one that solves the master problem, and one that solves the subproblem).

**Fig. 3** A general search procedure (algorithm) for solving a CSP

```

CSP_solver (Problem N)
  Create CSP_N For Problem N
  while not(solved or infeasible) do
    Remove_Inconsistent_Values      } %Constraint Propagation method
    Select_Decision_Variable         }
    Select_Value_For_Variable       } %Labeling
  end
  
```

Integration and hybridization of CLP and MP environments was also a subject of research conducted by the authors of this paper Sitek and Wikarek [20]; Sitek et al. [23]; Sitek and Wikarek [19]; Sitek and Wikarek [22]. Elimination and replacement of the variable distribution mechanism but retaining the mechanism of constraint propagation has become the key research direction. The introduction of transformation as a critical component of hybrid approach and backtracking from the distribution of variables in the CLP clearly distinguishes our approach from those known from the literature Bockmayr and Kasper [5]; Hooker [11]; Caricato and Grieco [7], Milano and Wallace [14].

In all our previous papers dealing with hybridization Sitek and Wikarek [20–22], the problem was modeled as a whole at one level, the model was transformed and on the basis of the post-transformation model, the ultimate problem in the form of MILP was generated. Even though the hybrid approach made it possible to solve larger problems and to reduce optimization time relative to mathematical programming methods, its effectiveness in these two areas was insufficient for SSCM (Sustainable Supply Chain Management) problems. This is how the idea of developing a multi-level approach implementing the UMSC concept appeared. The proposed solution was especially effective in solving discrete optimization problems for SSCM, as shown in the form of comparative analysis with the hybrid and MP methods in Fig. 13a, b and c. The contribution of this study is in the modification of the original hybrid approach and its extension to the multi-level form represented by a modified model transformed for the SSCM problem. This approach is able to use the ubiquitous modeling and solving constraints for combinatorial optimization problems in manufacturing and distribution problems. The proposed approach integrates CP, CLP, MP and Fuzzy Logic (FL). It is an extension of the hybrid approach but differs from it in two aspects. The problem has to be modeled in the form of a set of CSPs and new methodology of constraint solving (e.g. multiple presolving) has to be employed. In addition, the models in this paper, in the form of CSPs and CSPs<sup>T</sup> before and after transformation, were presented for SSCM problem for the first time. The corresponding constraints have a new, mostly binary, structure.

### 3 A multi-level approach to ubiquitous modeling and solving constraints for combinatorial optimization

The need for a convenient, efficient and effective way of modeling and solving various types of constraints was conducive to the development of the multi-level approach based on the results of the previous studies on

hybridization Sitek and Wikarek [20–22] and different, often complementary properties of individual paradigms (Section 2). The multi-level architecture results from the prevalence of constraints at various levels of the modeled problems. Figure 4 shows the conceptual diagram of the proposed approach.

The problem is modeled as a set of sub-problems in the form of CSPs for particular areas of production, distribution, storage, etc. (Level 1).

All CSPs are modeled using the set of CLP predicates. This set includes the predicates dedicated to a given CSP, e.g., predicates implementing individual constraints, general predicates used in each modeled problem, such as those creating lists of data from the sets of facts, and incorporated predicates (*disjunctive()*, *exclude()*, *cumulative()*, *alldifferent()* etc.). The modeling process uses the set of facts as an information layer for the problem.

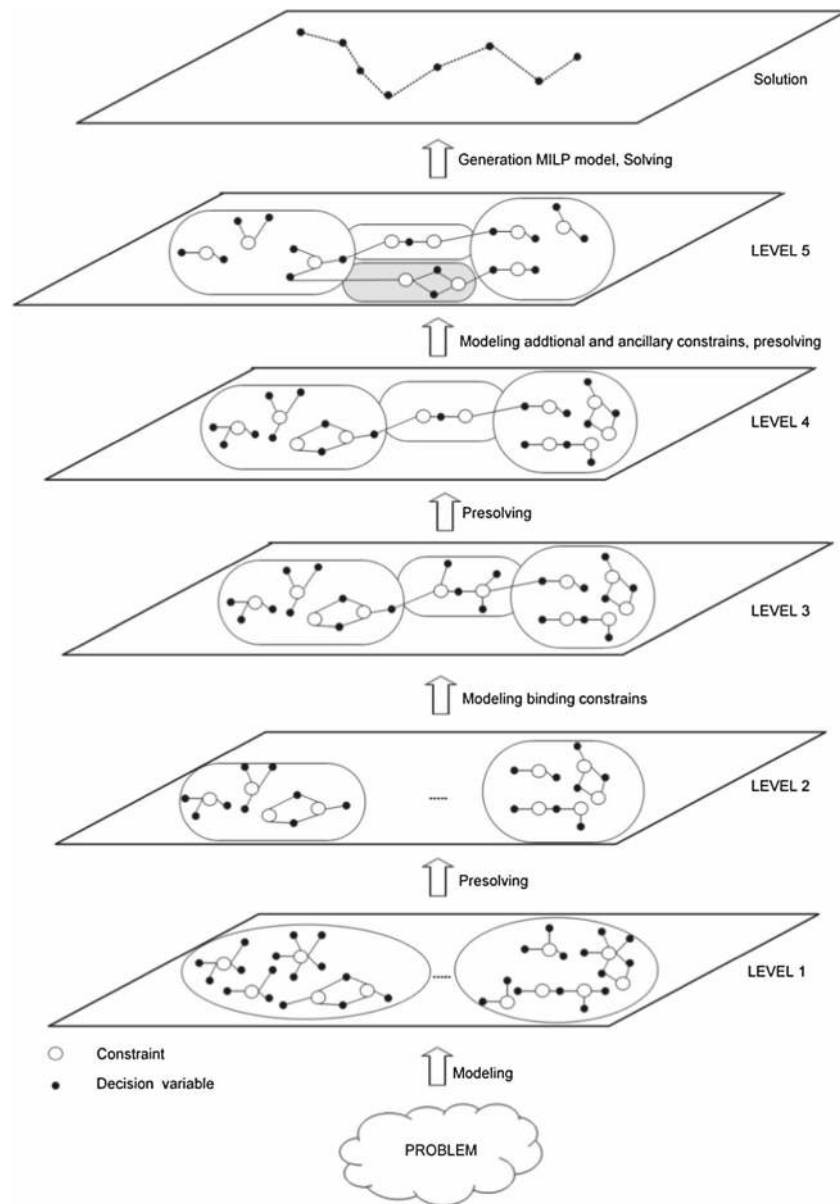
The CSP, standard or variant (DCSP, FCSP etc.) is then presolved (Level 2), using the methods which reduce the decision variable domains and can transform the variables and constraints. Constraint propagation, problem transformation or both methods combined into one are used as presolving methods (Section 3.1). In the next step, the constraints (financial, environmental or transport constraints) that bind the subproblems are modeled (Level 3). They are also presolved (Level 4). Now the additional and auxiliary constraints are modeled. These do not result directly from the structure of the problem but from the data instances, the user's specific requirements or possibility of increasing the effectiveness of the solution (Level 5). The next step involves presolving the model and generating the ultimate MP model. Finally, the MP model is optimized using the MP-based environment (Level 6). For illustrative example (Section 4) the MP model takes the form of MILP (Mixed Integer Linear Programming) model.

The multi-level architecture allows modeling complex problems through the parallel modeling of sub-problems at a given level and related constraints at subsequent levels. This will result in the multiple use of presolving thus increasing its effectiveness.

#### 3.1 Presolving

Presolving methods used at various levels constitute an important element of the proposed approach (Fig. 4). Presolving eliminates redundant information from the problem formulation while simultaneously trying to simplify and strengthen the formulation. It can be very effective and is often essential for solving instances. Especially for integer programming problems, fast and effective presolving methods are very important. Here, presolving may have a form of standard constraint propagation, problem transformation

**Fig. 4** The concept of multi-level approach to ubiquitous modeling and solving constraints

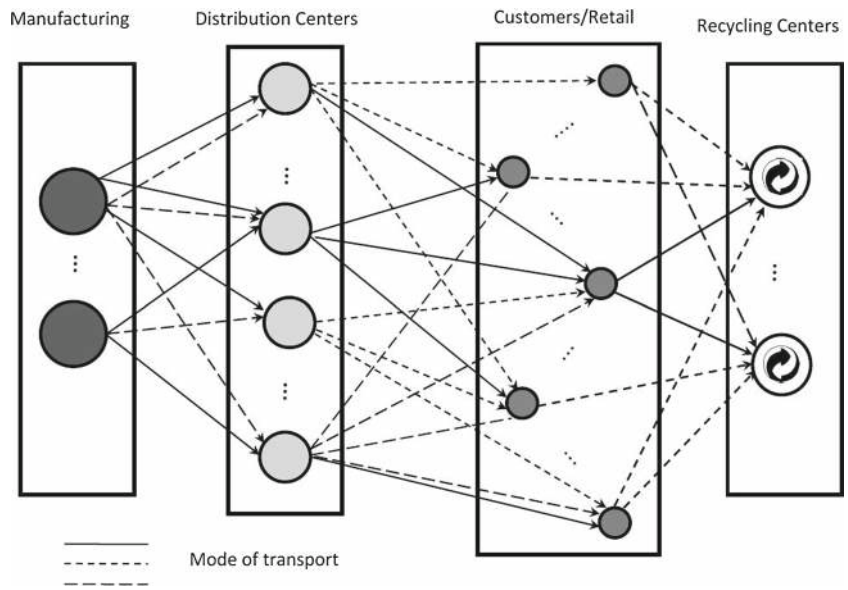


or a combination of these two methods. Constraint propagation is one of the CSP algorithm methods Rossi et al. [16]; Apt [1] whereas the transformation is the author's concept introduced, to varying degrees and in varying forms, to CP/MP environments hybridization Sitek and Wikarek [20, 22]; Sitek [18]; Wikarek [26].

In our approach, the transformation is the element that makes it possible to eliminate infeasible points in the decision variable space prior to defining the values of the decision variables. Typically, the transformation precedes the constraint propagation. The transformation method uses the set of data facts for describing the structure of a problem, and variable facts (orders, plans, etc). In this phase, the instances of facts are used in the analysis of relationships between data. The analysis allows

determining which connections or allocations are unacceptable/infeasible because no relevant/corresponding/adequate fact instances are available to describe them. As a result, adequate/corresponding decision variables will be removed in the presolving phase. For example, some of decision variables  $X_{f,c,t}$  that define the value of transport between factory  $f$  and distribution center  $c$  by transport means  $t$  will be removed from the model if a connection between  $f, c$  is impossible and/or a means of transport along the  $f, c$  route cannot be used. Therefore, let's do a simple analysis. Based on the illustrated example (Section 4, Appendix A), we know that  $f = 1..6$ ,  $c = 1..3$ ,  $t = 1..4$ . Thus the potential number of decision variables  $X_{f,c,t}$  is  $6 \times 3 \times 4 = 72$ . On the other hand, the number of instances of the fact  $fact_{f,c,t}()$  is 24 (Appendix A). In this case, the reduction involves

**Fig. 5** The simplified network of the supply chain



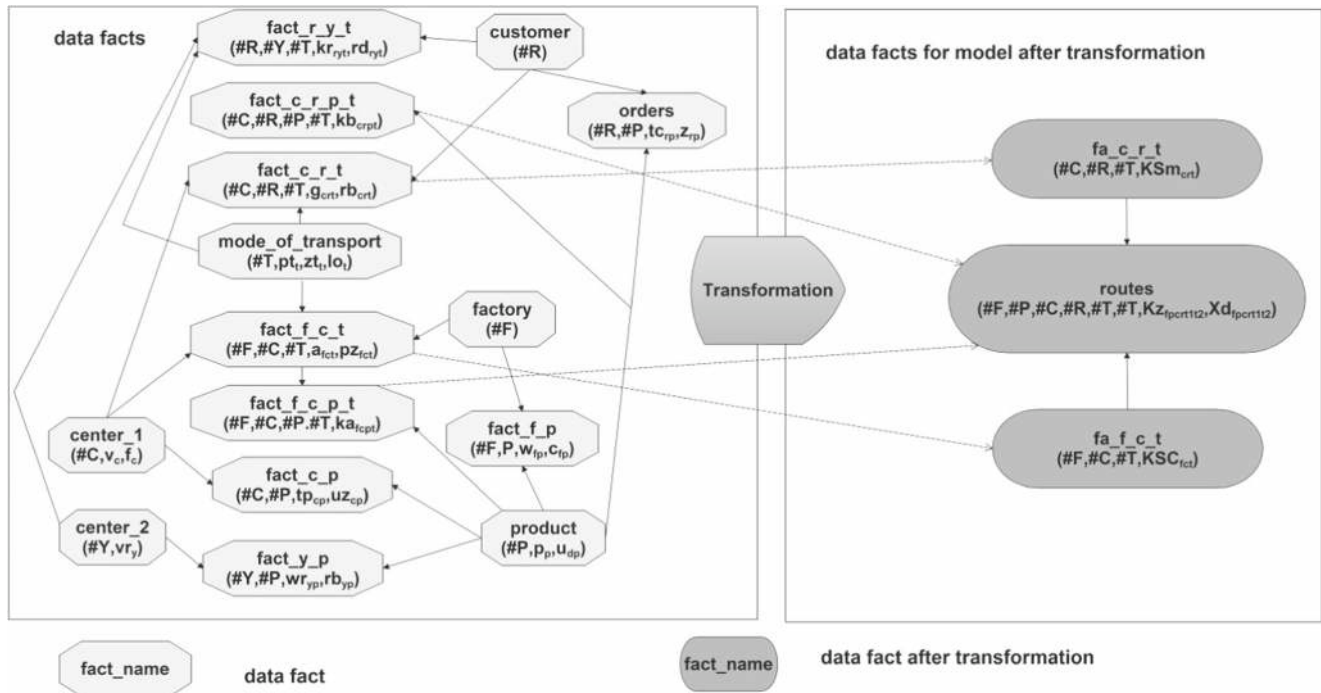
48 decision variables. So in the future it will be possible to define values only for 24 decision variables  $X_{f,c,t}$  for illustrative example (Figs. 5 and 6).

Likewise, the analysis of the remaining facts may result in the reduction of other decision variables of the modeled problem. In the transformed model, there will be much less decision variables than the initial one. Due to the multi-level hybrid approach, the reduction information will be transferred to the next level, resulting in further reductions

in both decision variables and constraints. The application of presolving in practice is shown in (Section 4).

### 3.2 Multi-level model

One of the distinguishing features of this multi-level approach is the method of modeling the problem. The model is represented as a set of CSPs with an architecture related to the modeled problem (Table 6 in Appendix D).



**Fig. 6** Diagram of relationships between facts for illustrative example before and after transformation (# - a key attribute for the fact)

Figure 7 shows a schematic diagram of the model together with possible architectures, depending on the problem characteristic features for illustrative example. The models can have a form of CSPs occurring in parallel at the same level, in series at different levels or in a mixed form as in the illustrative example (Section 4). Each CSP uses a set of facts as data and a set of questions (objectives) defining the way the constraints will be satisfied.

This way of modeling makes it very easy for a given class/type of problems, where changing sets of fact instances is all that has to be done. Modeling becomes more complicated in the case of a new problem class/type with new constraint types. These new constraints have to be modeled in the form of CLP predicates (Section 4).

For the illustrative example, completely new constraint formalization is proposed, resulting from the application of

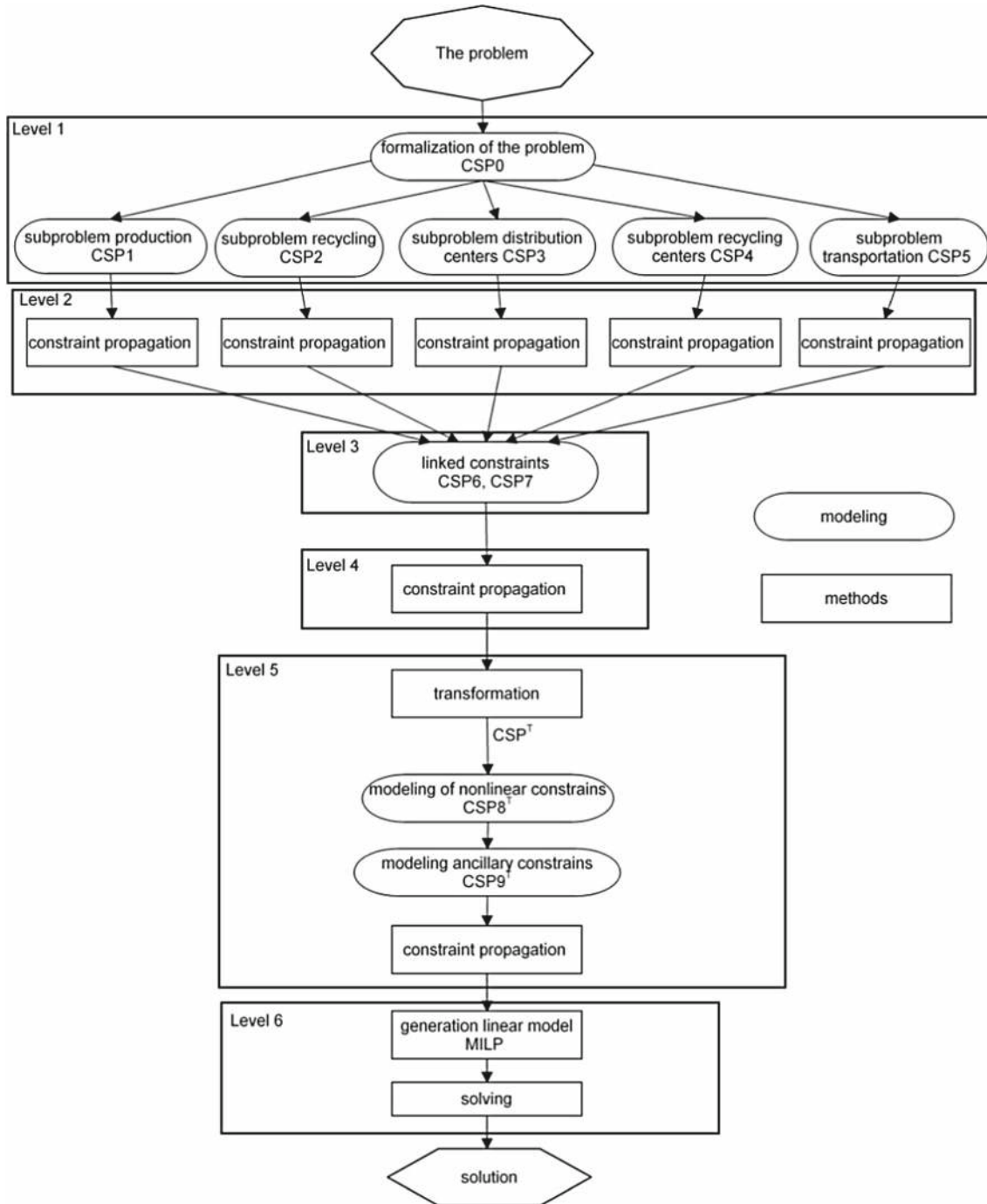


Fig. 7 The implementation framework for illustrative example based on the multi-level approach

the new multi-level architecture of the model. As a result, a large part of constraints had a binary form (Fig. 2), which is especially beneficial for the effectiveness of constraints propagation and transformation.

### 3.3 Innovativeness presented approach

Introducing presolving methods (constraint propagation and transformation) at different levels of problem modeling (Fig. 4) and mathematical programming solving methods is an important innovation of the proposed approach. Extremely time-consuming and generating a lot of back-trackings labeling is eliminated from the general search algorithm for solving CSPs (Fig. 3). Ubiquitous use of presolving methods in various forms and manners contributes substantially to the reduction, aggregation and transformation of decision variables and constraints. Thus the "lean" CSPs become the basis for generating a noticeably smaller MILP model.

For the solution of the generated model, we use mathematical programming methods and techniques instead of CLP, which is possible to avoid a lot of backtracking labeling techniques (extremely time-consuming). You can also skip other CLP techniques such as variable ordering, forward checking etc.

Another important novelty of the proposed approach is the ability to expand each of the subproblems modeled as CSPs without having to change and implement the whole problem as such Sitek and Wikarek [21, 22] This allows the use of a model with an architecture as presented in (Section 3.2). Knowledge of the problem is stored in the form of facts, which is possible due to the

declarative approach – CLP. Facts are also transformed (Fig. 6) and become a data layer for the model (3.2) (Appendix A, B).

### 4 Illustrative example

To show the practical application of the multi-level approach (Fig. 4), we used the problem of sustainable supply chain management (SSCM) Brandenburg and Rebs [6], with the formal model proposed in Sitek and Wikarek [19]. The problem, studied before, was chosen to compare the effectiveness of the proposed multi-layer approach with other solutions, in particular with our previous results Sitek and Wikarek [19], Sitek [19]. It is the combinatorial optimization problem which consists of the layers corresponding to production, distribution, recycling and retail. These layers are connected by transport (most frequently multimodal), as shown in Fig. 5. Each modeled layer has a specific set of constraints (mode of transport, production and distribution capacity, recycling centers capacity, etc.) presented in the form of interconnected CSPs. Such a method of modeling constraints essentially distinguishes the presented model from those published previously Sitek and Wikarek [21]; Sitek [19] where all constraints and objectives were modeled globally at the same time. The set of constraints that bind the layers (the number and capacity of transport means, environmental constraints, etc.) are also modeled as CSPs. The forms of CSPs for consecutive layers, the connecting, additional and logical constraints are shown in Table 1. The proposed way of problem modeling (Section 3.1) gave many of the constraints (1C1,

**Table 1** CSPs for illustrative example

CSP	Description
CSP0 presolving, provides binarity and integrality	$CSP0 = (C = \{1C0, 2C0\}), X = \{Z_{Xp}, X_{pf,p}, X_{ry,p}, X_{sc,p}, Tc_c, Trc_y, Xb_{f,c,t}, Yb_{c,r,t}, Zb_{r,y,t}, Xa_{f,c,p,t}, Ya_{c,r,p,t}, Xk_{f,c,p,t}, Yk_{c,r,p,t}, Zk_{j,y,p,t}\}$ $D = \{D_{Zx}^0, D_{Xp}^0, D_{Xr}^0, D_{Xs}^0, D_{Tc}^0, D_{Trc}^0, D_{Xb}^0, D_{Yb}^0, D_{Zb}^0, D_{Xa}^0, D_{Ya}^0, D_{Xk}^0, D_{Yk}^0, D_{Zk}^0\}$
CSP1 for production	$CSP1 = (C = \{1C1, 2C1, 3C1\}), X = \{X_{pf,p}\}, D = \{D_{Xp}^1\}$
CSP2 for recycling	$CSP2 = (C = \{1C2, 2C2, 3C2\}), X = \{X_{ry,p}\}, D = \{D_{Xr}^2\}$
CSP3 for distribution centers	$CSP3 = (C = \{1C3, 2C3, 3C3, 4C3\}), X = \{X_{sc,p}, Tc_c\}, D = \{D_{Xs}^3, D_{Tc}^3\}$
CSP4 for recycling centers	$CSP4 = (C = \{1C4, 2C4, 3C4, 4C4\}), X = \{X_{ry,p}, Trc_y\}, D = \{D_{Xr}^4, D_{Trc}^4\}$
CSP5 for transportation	$CSP5 = (C = \{1C5, 2C5, 3C5, 4C5\}), X = \{Xb_{f,c,t}, Yb_{c,r,t}, Zb_{r,y,t}\}, D = \{D_{Xb}^5, D_{Yb}^5, D_{Zb}^5\}$
CSP6 allows to link CSP5 with CSP1 and CSP2	$CSP6 = (C = \{1C6, 2C6, 3C6, 4C6, 5C6, 6C6, 7C6\}), X = \{Xb_{f,c,t}, Yb_{c,r,t}, Zb_{r,y,t}, Xa_{f,c,p,t}, Ya_{c,r,p,t}, Tc_c, Xk_{f,c,p,t}, Yk_{c,r,p,t}, Zk_{r,y,k,t}\}, D = \{D_{Xb}^6, D_{Yb}^6, D_{Zb}^6, D_{Xa}^6, D_{Ya}^6, D_{Tc}^6, D_{Xk}^6, D_{Yk}^6, D_{Zk}^6\}$
CSP7 allows to link CSP1 with CSP2, CSP3 and CSP4	$CSP7 = (C = \{1C7, 2C7, 3C7, 4C7, 5C7\}), X = \{Xk_{f,c,p,t}, Yk_{c,r,p,t}, Zk_{r,y,p,t}, X_{pf,p}, X_{sc,p}, X_{ry,p}\}, D = \{D_{Xk}^7, D_{Yk}^7, D_{Zk}^7, D_{Xp}^7, D_{Xs}^7, D_{Xr}^7\})n =$

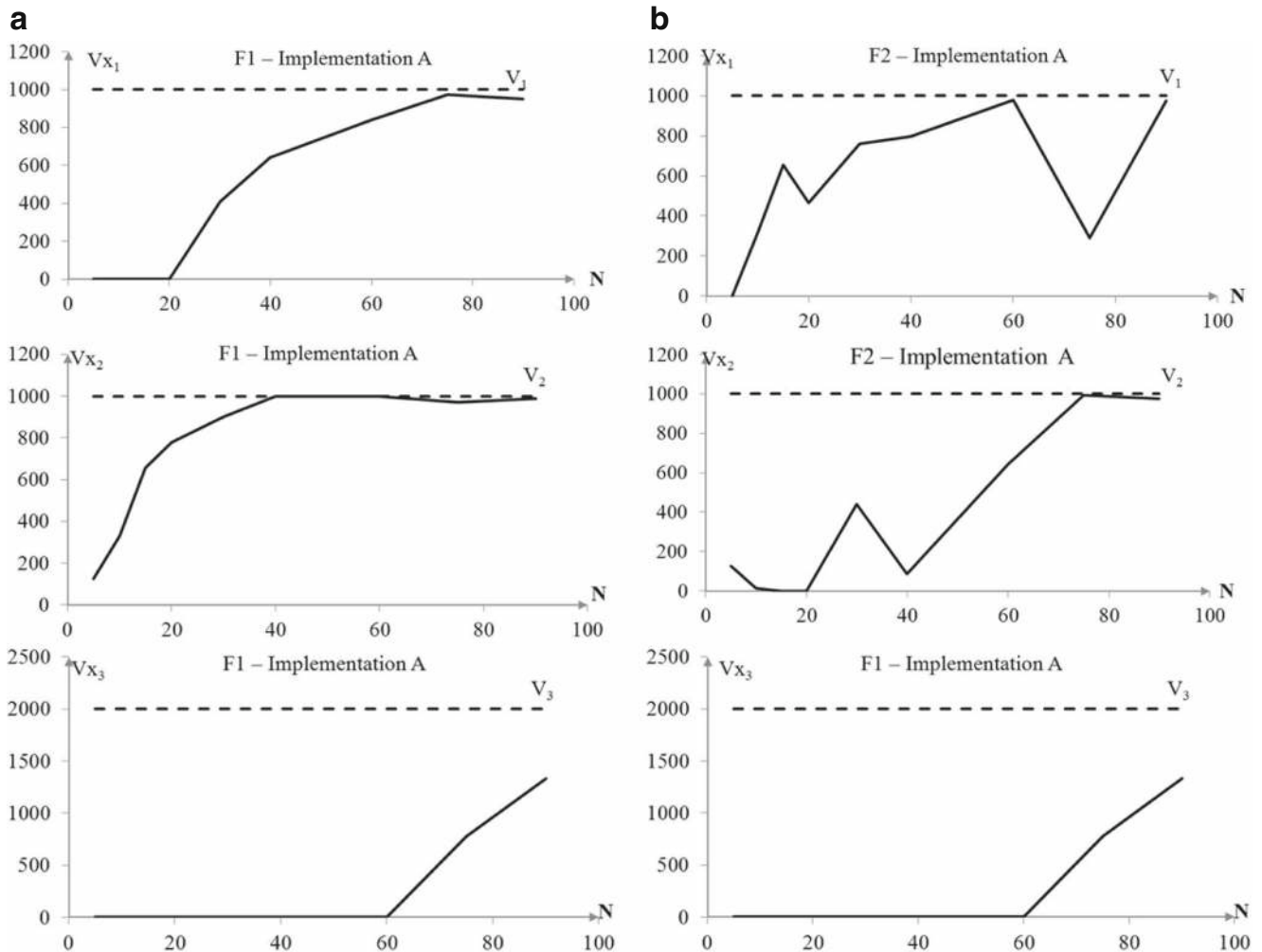


**Table 2** CSPs<sup>T</sup> for illustrative example

CSP	Description
CSP <sup>T</sup>	$CSP^T = (C = \{2Ta, 2Tb, 3T, 4T, 5Ta, 5Tb, 6T, 7T, 8T, 9T, 10T, 11T, 21Ta, 21Tb, 21Tc, 22T, 23T, 24T\}), X = \{X_{f,p,c,r,t,1,2}^T, Xb_{f,c,t}^T, Yb_{c,r,t}^T, Tc_c^T, Trc_c^T, Zr_{y,p,t}^T\}, D = \{D_{X^T}^0, D_{Xb^T}^0, D_{Yb^T}^0, D_{Tc^T}^0, D_{Trc^T}^0, D_{Zr^T}^0\}$
CSP8 <sup>T</sup> ensures the exclusion storage, transport and manufacturing	$CSP8^T = (C = \{26T, 27T, 28T\}), X = \{X_{f,p,c,r,d1,d2}^T, Zr_{r,y,k,d}^T\}, D = \{D_{X^T}^8, D_{Zr^T}^8\}$
CSP9 <sup>T</sup>	$CSP9^T = (C = \{12T, 13T, 14T, 15T, 16T, 17T, 18T, 19T, 20T\}), X = \{X_{f,p,c,r,d1,d2}^T, Xb_{f,c,t}^T, Yb_{c,r,t}^T\}, D = \{D_{X^T}^9, D_{Xb^T}^9, D_{Yb^T}^9\}$

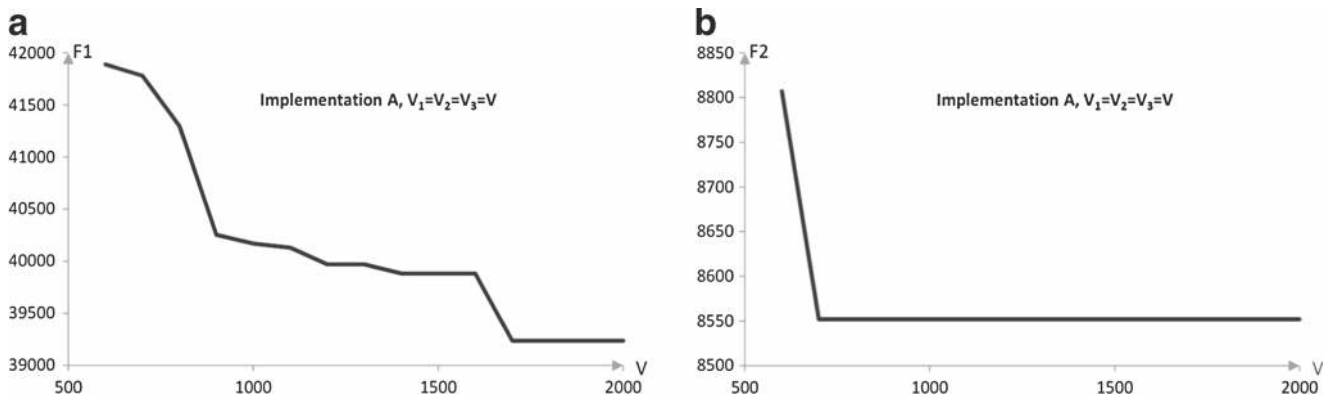
1C2, 2C2, 2C3, 2C4, 1C5, 2C5, 3C5, 1C6, 2C6, 3C6, see Appendix C and D) of the model (Table 1 and Appendix D) a binary character. This is another feature that differentiated SSCM problem formalization from other models

already presented in Sitek and Wikarek [21], Sitek [19]. The remaining constraints also have different forms. Thus, the proposed formalization (Appendix D) is a completely new concept.



**Fig. 8 a** Current use of distributors’ capacity ( $V_{x_1}, V_{x_2}, V_{x_3}$ ) for particular distributors depending on the numbers of orders ( $N$ ) (F1-objective function,  $V_1, V_2, V_3$ - distributors’ capacity). **b** Current use

of distributors’ capacity ( $V_{x_1}, V_{x_2}, V_{x_3}$ ) for particular distributors depending on the numbers of orders ( $N$ ) (F2-objective function,  $V_1, V_2, V_3$ - distributors’ capacity)



**Fig. 9** **a** Effect of the impact of capacity  $V$  (distributor) on the objective function  $F_c=F1$  ( $V=V_1=V_2=V_3$ ). **b** Effect of the impact of capacity  $V$  (distributor) on the objective function  $F_c=F2$  ( $V=V_1=V_2=V_3$ )

All the experiments and studies were carried out for the same data instances and in the same computational environment. Data instances for CSPs are presented using a set of facts. The structure of the facts describing the illustrative example, the relationship between facts and keys attributes for facts are shown in Fig. 6. The schematic diagram follows that well known from the database design, ERD (Entity Relationship Diagram) Teorey et al. [24]. The same diagram (Fig. 6) shows the transformation of facts for the illustrative example which is a presolving element and refers to the model data layer (Table 1 and Appendix A, Appendix B). Some of the facts after transformation change their structure and some are eliminated (15 facts before and 10 after transformation). It has to be noted that the number of instances of each fact is also greatly reduced.

Appendix A and B list the facts and decision variables for particular CSPs and CSPs<sup>T</sup>.

The implementation of the approach in Fig. 4, Section 3, which took the form of the framework, Fig. 7, was used to model and solve the illustrative example. Practical implementation of the UMSC concept in the framework form contains ubiquitous CSPs and ubiquitous presolving methods. Both elements are present at many levels (Fig. 7).

In this illustrative example, the presolving methods are represented by constraint propagation (Level 2, Level 4) and transformation with constraint propagation (Level 5). Constraint propagation reduces the domains of decision variables and, in some cases, detects the situations, in which certain decision variables do not satisfy the constraints and removes them. In our example, constraint propagation defined, among others, the minimum number of distribution and recycling centers needed for the processing of an order set and the minimum number of transport means needed for the distribution of products, etc. Transformation in this example included the change of problem representation by the elimination of infeasible routes, that is, the routes left out due to technological constraints or lack of orders for specific

products from specific distributors and manufacturers. The additional (CSP9<sup>T</sup>) and logical (CSP8<sup>T</sup>) constraints were modeled at the same level (Level 5). Table 2 shows the CSPs<sup>T</sup> after transformation. The transformation technique for the decision variables and constraints from the illustrative example is included in Appendix D (Tables 8 and 9). The Level 5 CSPs and selected objective functions (examples of objective functions are listed in Appendix C) were used to generate the problem formulated as an MILP model solved in the next step with an MP solver (Level 6). Full forms of CSPs and CSPs<sup>T</sup> with a description of the constraints are included in Appendix D (Tables 6 and 7).

An alternative approach consists of modeling the whole problem in the form of an MILP model that covers all the areas and all the constraints and solves the problem using the MP solver. This approach is less effective computationally, as only small size problems can be solved within acceptable time (Appendix E), with nonlinear and logical constraints excluded.

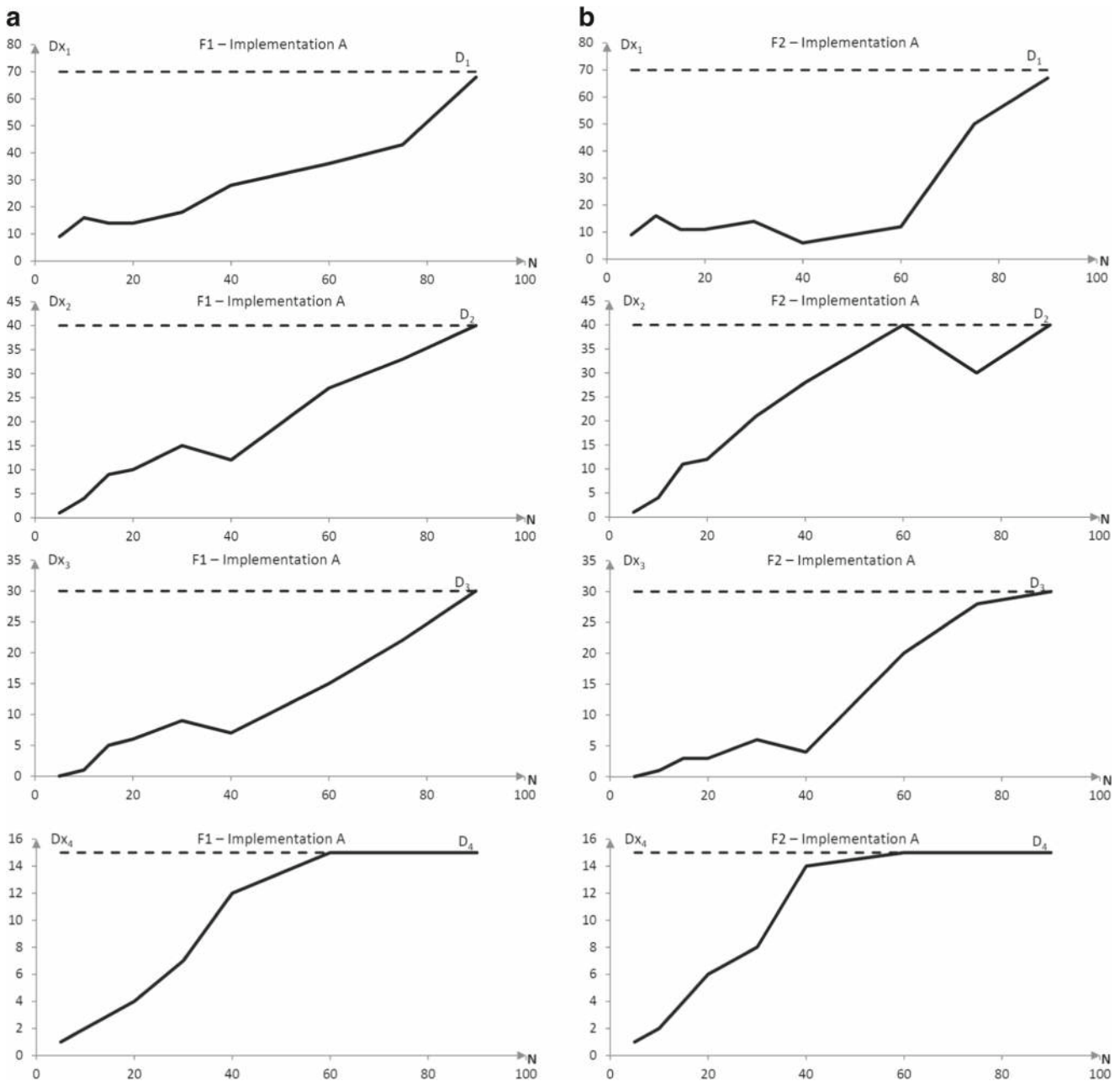
The tool of choice for the implementation of the framework (Fig. 7) was ECL<sup>i</sup>PS<sup>e</sup> Eclipse [8] which is an open-source software system for the cost-effective development and deployment of constraint programming applications. MP-based environment in the implementation framework was EPLEX built in ECL<sup>i</sup>PS<sup>e</sup> MP-solver. ECL<sup>i</sup>PS<sup>e</sup> was used to implement the levels of the framework: Level 1, Level 2, Level 3, Level 4 and Level 5. EPLEX was used for implementing Level 6 and for solving.

## 5 Computational experiments for illustrative example

In order to verify and evaluate the proposed framework, many computational experiments were performed for the illustrative example. All the experiments relate to the supply chain with six manufacturers ( $f=1..6$ ), three distributors

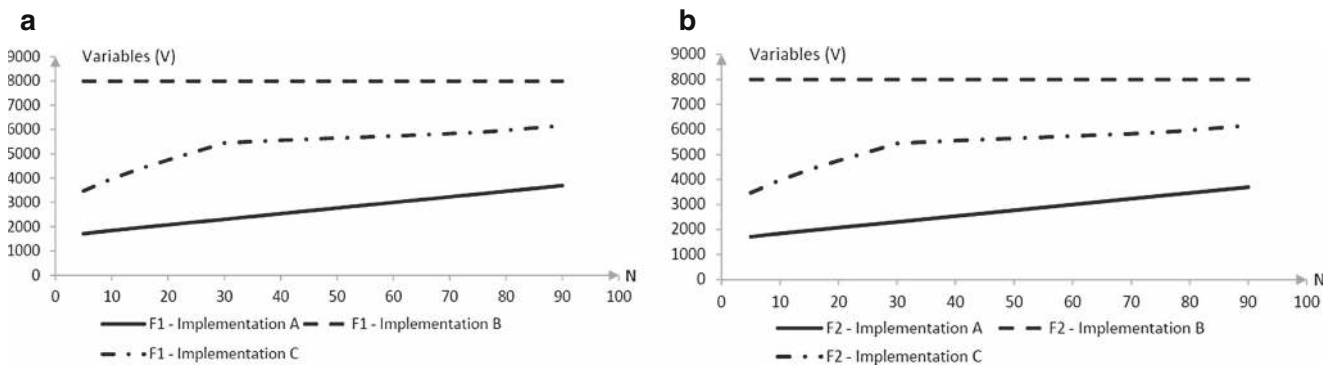
( $c = 1..3$ ), ten customers ( $r = 1..10$ ), four modes of transportation ( $t = 1..4$ ), two center<sub>2</sub> (recycling) ( $y = 1..2$ ), fifteen product items ( $p = 1..15$ ), and nine sets of orders ( $Or_1(5), Or_2(10), Or_3(15), Or_4(20), Or_5(30), Or_6(40), Or_7(60), Or_8(75), Or_9(90)$ )-(N)-the number of orders in set  $Or_i$ ). Two objective functions F1 (total supply chain costs) and F2 (total environmental costs) were used during the generation of the MILP model (Section 4, Fig. 7, Level 5). Both functions are shown in detail in Appendix C. The

experiments were carried out in two directions. Firstly, the methods for applying the proposed approach as a framework to the complex illustrative model (Implementation A) and examining the properties for the sample instance of data (Figs. 8a, b, 9a, b and 10a, b) are shown. At this stage, the values of  $Vx$ - current use of distributors' capacity,  $Dx$ - current use of transportation units were determined for different numbers of orders. The impact of parameter  $V$ - distributors' capacity on the objective function was



**Fig. 10 a** Current use of transportation units ( $Dx_1, Dx_2, Dx_3, Dx_4$ ) for different mode of transportation (F1-objective function,  $D_1, D_2, D_3, D_4$  - the number of transportation units for different mode of transportation) . **b** Current use of transportation units ( $Dx_1, Dx_2, Dx_3,$

$Dx_4$ ) for different mode of transportation (F2-objective function,  $D_1, D_2, D_3, D_4$  - the number of transportation units for different mode of transportation)

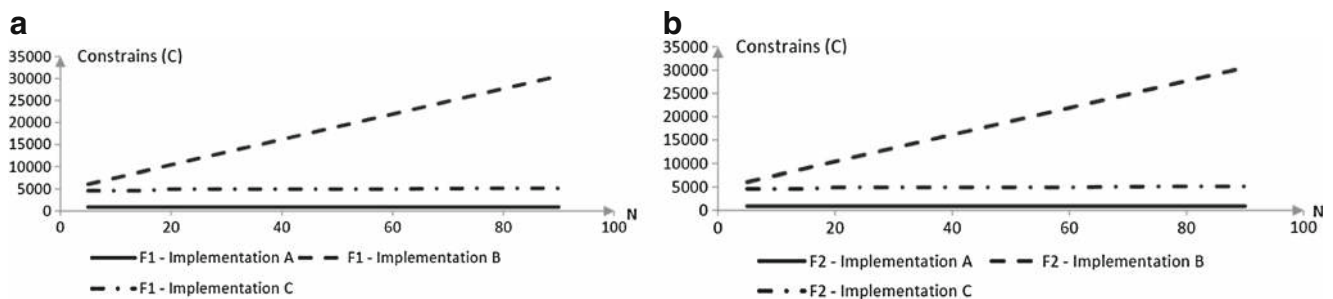


**Fig. 11** **a** Graph showing the number of decision variables of the implementation A, implementation B and implementation C for objective function F1. **b** Graph showing the number of decision variables of the implementation A, implementation B and implementation C for objective function F2

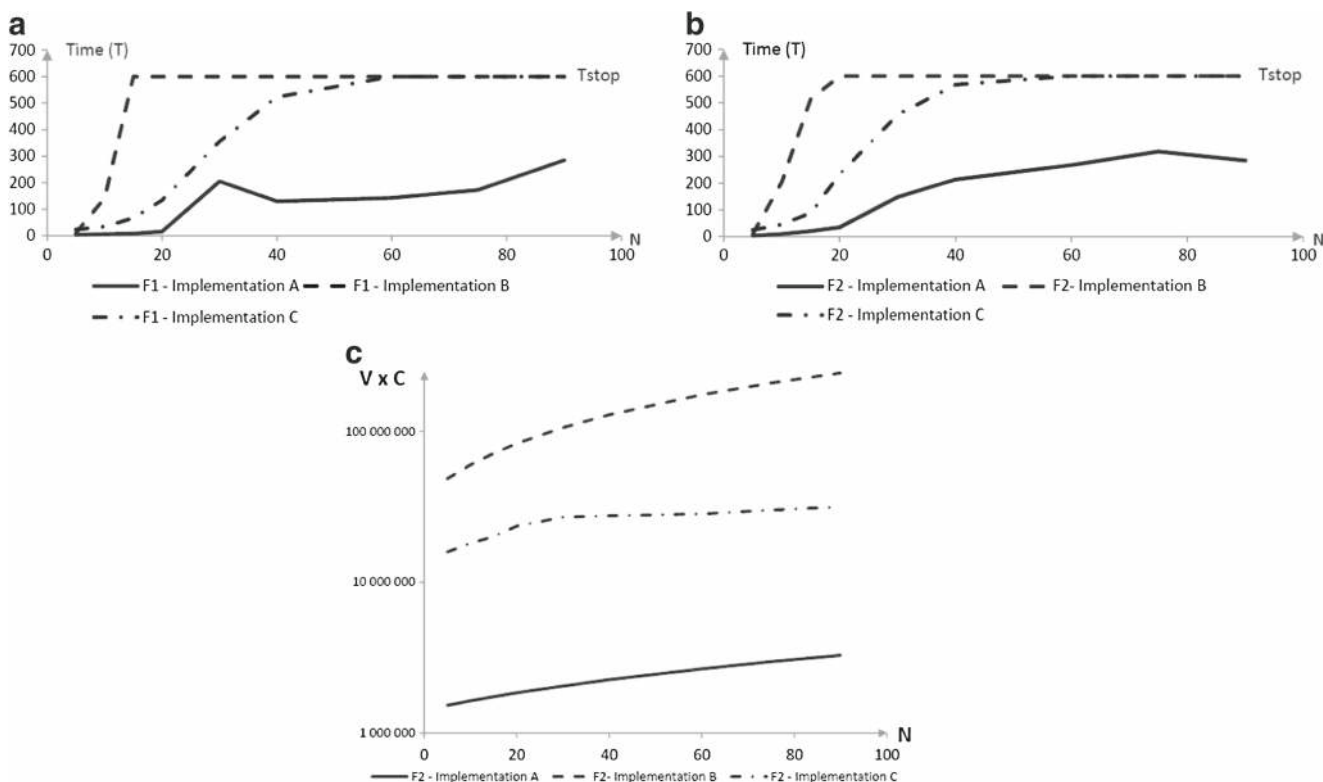
defined. The same procedure can be applied to evaluate the values and impacts of other model parameters. The information obtained from the evaluations forms the basis for decision support in SSCM. The key element of the proposed experiments was the consecutive stage in which effectiveness analysis for the proposed approach is performed in terms of computing time and problem size. For the comparative analysis, the same illustrative example was implemented in the MP-based approach and hybrid approach. Over 200 computational experiments were carried out for different model parameters and for nine series of data varied by the number of orders  $N$ . Analysis of the results of the second part of experiments clearly shows the benefit of the proposed approach (Implementation A) in relation to classical approaches based on mathematical programming (Implementation B) and hybrid approach (Implementation C). To ensure comparable conditions, all experiments were performed using a PC Intel core (TM2), 2.4 GHz, 2 GB RAM. The same MP solver was used in the same configuration in all three implementations (A,B,C).

Implementation A offers a 5-fold reduction in the number of variables (Fig. 11a, b), a 35-fold reduction in the number of constraints (Fig. 12a, b) and a 25-fold

reduction in the calculation time (Fig. 13a, b) compared with Implementation B. The search time was reduced when the data instances were small (up to 15 orders). In the case of larger size instances, the optimal solution could not be found with less than 600 s and the mathematical programming calculations were stopped (Appendix E). Compared with Implementation C, Implementation A offers a 3-fold reduction in the number of variables (Fig. 11a, b), a 6-fold reduction in the number of constraints (Fig. 12a, b) and a 6-fold reduction in the calculation time (Fig. 13a, b) for the same computational examples. The search time was reduced when the data instances were medium size (up to 60 orders). In the case of larger sizes of the data instances, the optimal solution could not be found with less than 600 s and the calculations were stopped (Appendix E). The results can be explained by analyzing potential solution space for each implementation. The space defined by the Cartesian product of  $V(\text{variables}) \times C(\text{Constraints})$  was determined for all numerical cases  $Or(N)$ . Figure 13c compares the size of these spaces for each implementation. Due to large differences between the sizes, the logarithmic scale was used. The differences between implementations reach at least an order of magnitude.



**Fig. 12** **a** Graph showing the number of constraints of the implementation A, implementation B and implementation C for objective function F1. **b** Graph showing the number of constraints of the implementation A, implementation B and implementation C for objective function F2



**Fig. 13** **a** Graph showing the calculation time of the implementation A, implementation B and implementation C for objective function F1 (Time stop=600s). **b** Graph showing the calculation time of the implementation A, implementation B and implementation C for objective

function F2 (Time stop=600s). **c** Graph showing the size (VxC) of the solution space for the implementation A, implementation B and implementation C (V-the number of decision variables, C-the number of constraints)

## 6 Conclusions

The paper provides a robust modeling and solving method to combinatorial optimization for production and distribution problems. This method, based on the concept of UMSC in multi-level architecture, is an extension to the hybrid approach, discussed in previous studies Sitek and Wikarek [20], Sitek and Wikarek [21], both in the area of modeling (modeling subproblems as separate CSPs, linked CSPs, logical CSPs and ancillary CSPs) and solving (with multiple use of presolving). The efficiency of the proposed method was examined through a set of computational experiments for illustrative example (optimization of SSCM). The experiments were also performed for the hybrid approach and mathematical programming. The outcome of the proposed method includes (a) flexible modeling and introduction of many binary constraints to the model, (b) a significant reduction in solution space, which allows solving larger problems in shorter times compared with hybrid approach and mathematical programming and (c) a possibility of independent modification and extension of the model CSPs.

The multi-level architecture is suitable for problems such as ubiquitous manufacturing, supply chain management,

distribution, vehicle routing etc., because it allows the implementation of constraints and presolving of the constraints in all areas and at all levels of the problem. This property is very important as it makes it possible to early detect constraint infeasibility in each subproblem. The use of the paradigms of constraint logic programming and mathematical programming in multi-level architecture takes advantage of the synergy of individual paradigms, which is clear in numerical experiments (Section 5 Fig. 11a and b, Appendix E Tables 10, 11, 12, 13, 14 and 15).

Further studies will focus on the implementation of other production-distribution models with the use of the proposed approach (Nielsen et al. [13]; Grzybowska and Gajšek [10]). The proposed method can be extended to modeling and solving soft and fuzzy constraints and multi-objective optimization. In the further extension of the method MP solvers will be replaced with heuristics and metaheuristics for solving industrial size problems. In future, a comparative analysis between the proposed approach and the pure CLP-based approach will be carried out. Advanced CLP techniques like different variable ordering techniques, GAC (Generalized-Arc-Consistency) for n-constraints will be tested in the context of modeling and solving distribution problems.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## Appendix A

The tables show the fact structure, including key attributes and parameters (Table 3), and decision variables (Table 4). All this information is needed to formalize the CSPs of the modeled problem. In brackets is the number of instances of a given fact, which are used for computational experiments (Section 4).

**Table 3** Summary facts and decision variables for CSPs

Facts	Keys	Parameters
Data facts (facts describing the structure of the problem)		
product (#p, parameters) [15]	#p -product item $p=1..P$ (P-number of products)	$P_p$ , - the volume/capacity occupied by product item p; $Ud_p$ -percentage coefficient of product item p
factory (#f, parameters) [6]	#f -factory $f=1..F$ (F-number of factories)	
center_2 (#y, parameters) [2]	#y - center_2 (recycling) $y=1..Y$ (Y-number of centers_2)	$Vr_y$ - center_2 y maximum capacity;
fact_f.p (#f,#p, parameters) [31]	#f - factory $f=1..F$ ; #p -product item $p=1..P$ ;	$W_{f,p}$ -production capacity at factory f for product item p; $C_{f,p}$ - the cost of product item p at factory f
center_1 (#c, parameters) [3]	#c - center_1(distribution) $c = 1..C$ (C- number of centers_1)	$V_c$ - center_1 c maximum capacity; $f_c$ - the fixed cost of center_1 c
fact_c.p (#c,#p, parameters) [40]	c - center_1(distribution) $c = 1..C$ #p - product item $p=1..P$	$Tp_{c,p}$ - the time needed for center_1 c to prepare the shipment of product item p; $Uz_{c,p}$ - if center_1 c can delivery product item p then $Uz_{c,p}=1$ , otherwise $Uz_{c,p}=0$
mode_of_transport (#t, parameters) [4]	#t - mode of transportation $t=1..T$ ; (T-number of mode of transportation)	$Pt_t$ the capacity of transportation unit; $Zt_t$ - the number of transportation units; $Lo_t$ - the environmental cost of using mode of transportation t;
fact_f.c.t (#f,#c,#t, parameters) [24]	#f - factory $i=1..F$ ; #c- center_1 $c = 1..C$ ; #t - mode of transportation $t=1..T$	$A_{f,c,t}$ -the fixed cost of delivery from factory f to center_1 c using mode of transportation t; $Pz_{f,c,t}$ - if factory f can delivery to center_1 c using mode of transportation t then $Pz_{f,c,t} = 1$ , otherwise $Pz_{f,c,t} = 0$
fact_y.p (#y,#p, parameters) [25]	#y - center_2 $y=1..Y$ ; #p - product item $p=1..P$	$Wr_{y,p}$ -capacity at center_2 for product item p; $Rb_{y,p}$ - if recycling center_2 can utilized product p then $Rb_{y,p} = 1$ , otherwise $Rb_{y,p} = 0$ (for MILP models)
customer(#r) [10]	#r - customer $r=1..R$ ; (R- number of customers)	
fact_c.r.t (#c,#r,#t, parameters) [60]	#c - center_1 $c=1..C$ ; #r - customer $r=1..R$ ; #t - mode of transportation $t=1..T$	$G_{c,r,t}$ -the fixed cost of delivery from center_1 c to customer r using mode of transportation t; $Rb_{c,r,t}$ - if center_1 c can delivery to customer r using mode of transportation t then $Rb_{c,r,t} = 1$ , otherwise $Rb_{c,r,t} = 0$ (for MILP models)
fact_r.y.t (#r,#y,#t, parameters) [45]	#r - customer $r=1..R$ ; #y - center_2 $y=1..Y$ ; #t - mode of transportation $t=1..T$	$Kr_{r,y,t}$ - the fixed cost of delivery from customer r to center_2 y using mode of transportation t; $Rd_{r,y,t}$ - if customer r can deliver to center_2 y using mode of transportation t then $Rd_{r,y,t} = 1$ , otherwise $Rd_{r,y,t} = 0$ (for MILP models)
fact_f.c.p.t (#f,#c,#p,#t, parameters) [360]	#f - factory $f=1..F$ ; #c - center_1 $c=1..C$ ; #p - product item $p=1..P$ ; #t - mode of transportation $t=1..T$	$Ka_{f,c,p,t}$ - the variable cost of delivery of product item p from factory f to center_1 c using of mode of transportation t
fact_c.r.p.t (#c,#r,#p,#t, parameters)[800]	#c - center_1 $c=1..C$ ; #r - customer $r=1..R$ ; #p - product item $p=1..P$ ; #t - mode of transportation $t=1..T$	$Kb_{c,r,p,t}$ - the variable cost of delivery of product item p from center_1 c to customer r using mode of transportation t

**Table 3** (continued)**Input facts**orders ( $\#r, \#p$ , parameters) [max 90]# $p$  - product item  $p=1..P$ ; # $r$ -  
customer  $r=1..R$  $T_{c,r,p}$  - the cut-off time of delivery to the delivery point/customer  $r$  of product item  $p$ ;  $Z_{r,p}$  - customer demand/order  $r$  for product item  $p$ **Decision variables** $X_{pf,p}$ the volume of production item  $p$  in the factory  $f$  $X_{ry,p}$ the product item  $p$  rate in a center  $y$  $X_{sc,p}$ the number of item  $p$  that is handled at the center  $c$  $T_{c_c}$ if center  $c$  participates in deliveries, then  $T_{c_c} = 1$ , otherwise  $T_{c_c} = 0$  $Tr_{c_y}$ if center  $y$  participates in utilization, then  $Tr_{c_y} = 1$ , otherwise  $Tr_{c_y} = 0$  $X_{bf,c,t}$ the number of courses from factory  $f$  to center  $c$  using mode of transportation  $t$  $Y_{bc,r,t}$ the number of courses from center  $c$  to customer  $r$  using mode of transportation  $t$  $Z_{br,y,t}$ the number of courses from customer  $r$  to center  $y$  using mode of transportation  $t$  $XK_{f,c,p,t}$ the size of delivery item  $p$  from factory  $f$  to center  $c$  using mode of transportation  $t$  $YK_{c,r,p,t}$ the size of delivery item  $p$  from center  $c$  to customer  $r$  using mode of transportation  $t$  $ZK_{r,y,p,t}$ the size of delivery item  $p$  from customer  $r$  to center  $y$  using mode of transportation  $t$  $Xa_{f,c,p,t}$ if delivery item  $p$  is from factory  $f$  to center  $c$  using mode of transportation  $t$  then  $Xa_{f,c,p,t} = 1$ , otherwise  $Xa_{f,c,p,t} = 0$  $Za_{r,y,t}$ if delivery is from customer  $r$  to center  $y$  using mode of transportation  $t$  then  $Za_{r,y,t} = 1$ , otherwise  $Za_{r,y,t} = 0$  $Ya_{c,r,p,t}$ if delivery product item  $p$  is from center  $c$  to customer  $r$  using mode of transportation  $t$  then  $Ya_{c,r,p,t} = 1$ , otherwise  $Ya_{c,r,p,t} = 0$ **Values calculated** $ZX_p$ cumulative product item  $p$  demand $Ko_{af,c,t}$ the total cost of delivery from factory  $f$  to center  $c$  using mode of transportation  $t$  $Vx_c$ the value corresponds to the center  $c$  real uptake capacity $Kog_{c,r,t}$ the total cost of delivery from center  $c$  to customer  $r$  using mode of transportation  $t$  $Cw$ 

arbitrarily large constant, for instance, the sum of all orders

**Appendix B**

The tables show the fact structure, including key attributes and parameters (Table 4), and decision variables (Table 5)

after transformation. All this information is needed to formalize the CSP<sup>T</sup>s of the modeled problem. In brackets is the number of instances of a given fact, which are used for computational experiments (Section 4).

**Table 4** Summary facts for CSPs<sup>T</sup>

Facts	Keys	Parameters
<b>Data facts after transformation (facts describing the structure of the problem)</b>		
routes (#f,#p,#c,#r,#t1,#t2, parameters) [2076]	#f - factory $f = 1..F$ ; #p -product item $p = 1..P$ #c - center_1 $c=1..C$ ; #r - customer $r = 1..R$ ; #t1 - mode of transportation $t1=1..T$ #t2 - mode of transportation $t2 = 1..T$	$Kz_{f,p,c,r,t1,t2}$ -the cost of delivery of product item $p$ from factory $f$ to customer $r$ via center_1 $c$ using of mode of transportation $t1$ and $t2$ $Xd_{f,p,c,r,t1,t2}$ – if available delivery of product item $p$ from factory $f$ to customer $r$ via center_1 $c$ using of mode of transportation $t1$ and $t2$ then $Xd_{f,p,c,r,t1,t2} = 1$ otherwise $Xd_{f,p,c,r,t1,t2} = 0$
fa_c_r_t (#c,#r,#t,parameters) [60]	#c - center_1 $c = 1..C$ ; #r - customer $r = 1..R$ ; #t - mode of transportation $t = 1..T$	$KSm_{c,r,t}$ - the fixed cost of delivery from center_1 $c$ to customer $r$ using mode of transportation $t$ ;
fa_f_c_t (#f,#c,#t,parameters) [24]	#f - factory $f = 1..F$ ; #c - center_1 $c = 1..C$ ; #t - mode of transportation $t = 1..T$	$KSC_{f,c,t}$ - the fixed cost of delivery from factory $f$ to center_1 $c$ using mode of transportation $t$ ;
<b>Data facts without transformation</b>		
product(#p, parameters), fact_f_p(#f,#p, parameters), center_1 (#c, parameters) center_2 (#y, parameters) mode_of_transport(#t, parameters), fact_y_p(#y,#p, parameters), orders(#r,#p, parameters).		

**Table 5** Summary indices,parameters and decision variables for CSPs<sup>T</sup>

<b>Input parameters</b>	
$Rmin_t$	the minimum number of means of transport $t$ (determines presolving)
$Rmax_t$	the maximum number of means of transport $t$ (determines presolving)
$Fc$	the minimum number of means of transport on the route from factory to center_1 (determines presolving)
$Cm$	the minimum number of means of transport on the route from center_1 to customer (determines presolving)
$Sro$	the minimum number of means of transport (determines presolving)
$Kz_{f,p,c,r,t1,t2}$	the variable cost of delivery of product item $p$ from factory $f$ to customer $r$ via center_1 $c$ using mode of transport $t1$ and $t2$ (determines presolving)
$Xd_{f,p,c,r,t1,t2}$	if possible delivery of product item $p$ from factory $f$ to customer $r$ via center_1 $c$ using mode of transport $t1$ and $t2$ than $Xd_{f,p,c,r,t1,t2} = 1$ otherwise $Xd_{f,p,c,r,t1,t2} = 0$ (determines presolving)
$Ksc_{f,c,t}$	the total cost of delivery from factory $f$ to center_1 $c$ using mode of transportation $t$ (determines presolving)
$Ksm_{c,r,t}$	the total cost of delivery from center_1 $c$ to customer $r$ using mode of transportation $t$ (determines presolving)
<b>Decision variables</b>	
$X_{f,p,c,r,t1,t2}^T$	the size of delivery item $p$ from factory $f$ to customer $r$ via center_1 $c$ using mode of transport $t1$ and $t2$
$Xb_{f,c,t}^T$	the number of courses from factory $f$ to center_1 $c$ using mode of transportation $t$
$Yb_{c,r,t}^T$	the number of courses from center_1 $c$ to customer $r$ using mode of transportation $t$
$Tc_c^T$	if center_1 $c$ participates in deliveries, then $Tc_c = 1$ , otherwise $Tc_c = 0$
$Tb_y^T$	if center_2 $y$ participates in deliveries, then $Tb_y = 1$ , otherwise $Tb_y = 0$
$Zr_{r,y,p,t}^T$	the size of delivery item $p$ from customer $r$ to center_2 $y$ using mode of transportation $t$
$Za_{r,y,t}^T$	if delivery is from customer $r$ to center_2 $y$ using mode of transportation $t$ then $Za_{r,y,t}^T = 1$ , otherwise $Za_{r,y,t}^T = 0$
$Zb_{r,y,t}^T$	the number of courses from customer $r$ to center_2 $y$ using mode of transportation $t$



### Appendix C

$$\begin{aligned}
 F1 &= \sum_{c=1}^C (F_c \cdot Tc_c^T) + \sum_{t=1}^T Lo_t \left( \sum_{f=1}^F \sum_{c=1}^C Xb_{f,c,t}^T + \sum_{c=1}^C \sum_{r=1}^R Yb_{c,r,t}^T \right) + \sum_{f=1}^F \sum_{c=1}^C \sum_{t=1}^T Koaf_{c,t} + \sum_{c=1}^C \sum_{r=1}^R \sum_{t=1}^T Kog_{c,r,t} + \sum_{y=1}^Y (Fr_y \cdot Tb_y^T) + \\
 &\sum_{f=1}^F \sum_{p=1}^P (C_{f,p} \cdot \sum_{c=1}^C \sum_{r=1}^R \sum_{t=1}^T \sum_{t2=1}^T X_{f,p,c,r,t,t2}^T) - \sum_{y=1}^Y \sum_{p=1}^P (Cr_{y,p} \cdot \sum_{r=1}^R \sum_{t=1}^T Z_{r,y,p,t}^T) + \sum_{t=1}^T Lo_t \left( \sum_{y=1}^Y \sum_{r=1}^R Zb_{r,y,t}^T \right) + \sum_{r=1}^R \sum_{y=1}^Y \sum_{t=1}^T Kr_{r,y,t} * Zb_{r,y,t}^T \\
 F2 &= \sum_{t=1}^T Lo_t \left( \sum_{f=1}^F \sum_{c=1}^C Xb_{f,c,t}^T + \sum_{c=1}^C \sum_{r=1}^R Yb_{c,r,t}^T \right) - \sum_{y=1}^Y \sum_{p=1}^P (Cr_{y,p} \cdot \sum_{r=1}^R \sum_{t=1}^T Z_{r,y,p,t}^T) + \sum_{t=1}^T Lo_t \left( \sum_{y=1}^Y \sum_{r=1}^R Zb_{r,y,t}^T \right)
 \end{aligned}$$

F1- total supply chain costs minus the profit from recycling  
 F2-environmental total supply chain costs minus the profit from recycling.

Table 6 shows the explicit form of CSPs. For each CSP the set of constraints is presented in algebraic form.

Analogously, Table 7 shows the explicit form of CSPs<sup>T</sup> (after transformation). Comparison of decision variables and constraints before and after transformation are presented in Tables 8 and 9.

### Appendix D

**Table 6** Explicit forms of CSPs with a description of the constraints

CSPs	Constrains
	$\sum_{r=1}^R Z_{r,p} = Zx_p \quad \forall p = 1..P \tag{1C0}$
CSP0	Determining the cumulative demand for particular products $p$ . $X_{pf,p} \in N \quad \forall f = 1..F, p = 1..P \tag{2C0}$ $X_{ry,p} \in N \quad \forall y = 1..Y, p = 1..P$ $X_{sc,p} \in N \quad \forall c = 1..C, p = 1..P$ $Tc_c \in \{0, 1\} \quad \forall c = 1..C$ $Tr_{cy} \in \{0, 1\} \quad \forall y = 1..Y$ $Xb_{f,c,t} \in N \quad \forall f = 1..F, c = 1..C, t = 1..T$ $Xaf_{c,p,t} \in \{0, 1\} \quad \forall f = 1..F, c = 1..C, p = 1..P, t = 1..T$ $Yb_{c,r,t} \in N \quad \forall c = 1..C, r = 1..R, t = 1..T$ $Yac_{r,p,t} \in \{0, 1\} \quad \forall c = 1..C, r = 1..R, p = 1..P, t = 1..T$ $Zb_{b,j,d} \in N \quad \forall b = 1..U, j = 1..M, d = 1..L$ $Xkf_{c,p,t} \in N \quad \forall f = 1..F, c = 1..C, p = 1..P, t = 1..T$ $Ykc_{r,p,t} \in N \quad \forall c = 1..C, r = 1..R, p = 1..P, t = 1..T$ $Zkr_{y,p,t} \in N \quad \forall r = 1..R, y = 1..Y, p = 1..P, t = 1..T$
CSP1	Integrity and binarity $X_{pf,p} \leq W_{f,p} \quad \forall f = 1..F, p = 1..P \tag{1C1}$ The production capacity $W_{f,p}$ of the factory is not exceeded. $\sum_{p=1}^P \frac{X_{pf,p}}{W_{f,p}} \leq 1 \quad \forall f = 1..F \tag{2C1}$ The joint production capacity in the factory $f$ for all item $p$ . $\sum_{f=1}^F X_{pf,p} = Zx_p \quad \forall p = 1..P \tag{3C1}$ Production covers all customer orders.

**Table 6** (continued)

<p>CSP2 <math>ZX_{r,y,p} \leq W_{r,y,p} \quad \forall y = 1..Y, p = 1..P</math> (1C2)</p> <p>The capacity of the recycling center will not be exceeded.</p> <p><math>X_{r,y,p} \leq W_{r,y,p} \quad \forall y = 1..Y, p = 1..P</math> (2C2)</p> <p>The joint capacity in the recycling center <math>y</math> for all item <math>p</math>.</p> <p><math>\sum_{y=1}^Y X_{r,y,p} \cdot R_{b,y,p} = U_{d,p} \cdot Z_{X_p} \quad \forall p = 1..P</math> (3C2)</p> <p>The percentage of sold item <math>p</math> that is recycled.</p>	<p>CSP3 <math>\sum_{c=1}^C X_{s_c,p} = Z_{X_p} \quad \forall p = 1..P</math> (1C3)</p> <p>The quantity of item <math>p</math> will be handled in the distribution center <math>c</math> which result from the customer orders.</p> <p><math>X_{s_c,p} \cdot P_p \cdot U_{z_c,p} \leq V_c \quad \forall c = 1..C, p = 1..P</math> (2C3)</p> <p>Distribution only specific items <math>p</math> by the distribution center <math>c</math> due to technological conditions.</p> <p><math>\sum_{p=1}^P (X_{s_c,p} \cdot P_p) \leq V_c \quad \forall c = 1..C</math> (3C3)</p> <p>The total volume of items <math>p</math> distributed by each distribution center <math>c</math> is less than the capacity of this center.</p> <p><math>\sum_{c=1}^C \sum_{p=1}^P (X_{s_c,p} \cdot P_p \cdot U_{z_c,p}) \leq \sum_{c=1}^C V_c \cdot T_{c_c}</math> (4C3)</p> <p>The available capacity of distribution centers <math>c</math> is greater than cumulative capacity of distributed items <math>p</math>.</p>
<p>CSP4 <math>\sum_{y=1}^Y X_{r,y,p} = U_{d,p} \cdot Z_{X_p} \quad \forall p = 1..P</math> (1C4)</p> <p>The recycling center <math>y</math> will be recycled only those items <math>p</math> which enables its technology.</p> <p><math>X_{r,y,p} \cdot P_p \leq R_{r,y,p} \cdot V_{r,y} \quad \forall y = 1..Y, p = 1..P</math> (2C4)</p> <p>The available capacity of recycling centers <math>y</math> is greater than the total volume of recyclable items <math>p</math>.</p> <p><math>\sum_{y=1}^Y (X_{r,y,p} \cdot P_p) \leq R_{r,y,p} \cdot V_{r,y} \quad \forall y = 1..Y</math> (3C4)</p> <p>The cumulative volume of recyclable items <math>p</math> for each recycling center <math>y</math> is less than the capacity of this center.</p> <p><math>\sum_{y=1}^Y \sum_{p=1}^P (X_{r,y,p} \cdot P_p) \leq \sum_{y=1}^Y R_{r,y,p} \cdot V_{r,y} \cdot T_{r_c y}</math> (4C4)</p> <p>Cumulative available capacity of recycling centers <math>y</math> is greater than cumulative capacity of recycled items <math>p</math>.</p>	<p>CSP5 <math>X_{b_{f,c,t}} \leq P_{z_{f,c,t}} \cdot Z_{t_t} \quad \forall f = 1..F, c = 1..C, t = 1..T</math> (1C5)</p> <p>The deliveries from the factory <math>f</math> to the distribution center <math>c</math> are transported along the existing routes by acceptable means of transport <math>t</math>.</p> <p><math>Y_{b_{c,r,t}} \leq R_{b_{c,r,t}} \cdot Z_{t_t} \quad \forall c = 1..C, r = 1..R, t = 1..T</math> (2C5)</p> <p>The deliveries from the distribution center <math>c</math> to the customer <math>r</math> are transported along the existing routes by acceptable means of transport <math>t</math></p> <p><math>Z_{b_{r,y,t}} \leq R_{d_{r,y,t}} \cdot Z_{t_t} \quad \forall r = 1..R, y = 1..Y, t = 1..T</math> (3C5)</p> <p>The deliveries from the customer <math>r</math> to the recycling center <math>y</math> are made along the existing routes by acceptable means of transport <math>t</math>.</p> <p><math>\sum_{f=1}^F \sum_{c=1}^C X_{b_{f,c,t}} + \sum_{c=1}^C \sum_{r=1}^R Y_{b_{c,r,t}} + \sum_{r=1}^R \sum_{y=1}^Y Z_{b_{r,y,t}} \leq \cdot Z_{t_t} \quad \forall t = 1..T</math> (4C5)</p> <p>Transportation process does not use more means of transport <math>t</math> than is currently available.</p>
<p>CSP6 <math>X_{b_{f,c,t}} \leq C_w \cdot T_{c_c} \quad \forall f = 1..F, c = 1..C, t = 1..T</math> (1C6)</p> <p><math>X_{b_{f,c,t}} \geq X_{a_{f,c,p,t}} \quad \forall f = 1..F, c = 1..C, p = 1..P, t = 1..T</math></p> <p>Transportation process from the factory <math>f</math> to the operating distribution centers <math>c</math> using acceptable means of transport <math>t</math>.</p> <p><math>Y_{b_{c,r,t}} \leq C_w \cdot T_{c_c} \quad \forall c = 1..C, r = 1..R, t = 1..T</math> (2C6)</p> <p><math>Y_{b_{c,r,t}} \geq Y_{a_{c,r,p,t}} \quad \forall c = 1..C, r = 1..R, p = 1..P, t = 1..T</math></p> <p>Transportation process from operating distribution centers <math>c</math> to the customer <math>r</math> using acceptable means of transport <math>t</math>.</p> <p><math>Z_{b_{r,y,t}} \leq C_w \cdot T_{r_c y} \quad \forall r = 1..R, y = 1..Y, t = 1..T</math> (3C6)</p> <p>Transportation process from the customer <math>r</math> to the operating recycling centers <math>y</math>.</p> <p><math>\sum_{p=1}^P (X_{k_{f,c,p,t}} \cdot P_p) \leq X_{b_{f,c,t}} \cdot P_{t_t} \quad \forall f = 1..F, c = 1..C, t = 1..T</math> (4C6)</p> <p>Constraint sets the adequate number of transport means <math>t</math> for transporting items <math>p</math> from factory <math>f</math> to distribution center <math>c</math>.</p> <p><math>\sum_{p=1}^P (Y_{k_{c,r,p,t}} \cdot P_p) \leq Y_{b_{c,r,t}} \cdot P_{t_t} \quad \forall c = 1..C, r = 1..R, t = 1..T</math> (5C6)</p>	<p>(5C6)</p>

**Table 6** (continued)

	Constraint sets the adequate number of transport means $t$ for transporting items $p$ from distribution center $c$ to customer $r$ .	
	$\sum_{p=1}^P (Zk_{r,y,p,t} \cdot P_p) \leq Zb_{r,y,t} \cdot Pt_t \quad \forall r = 1..R, y = 1..Y, t = 1..T$	(6C6)
	Constraint sets the adequate number of transport means $t$ for transporting items $p$ from customer $r$ to recycling centery.	
	$Tf_{f,c,t1} \cdot Xa_{f,c,p,t1} + Xa_{f,c,p,t1} \cdot Tp_{c,p} + Tm_{c,r,t2} \cdot Ya_{c,r,t2} \leq Tc_{r,p} \quad \forall i = f..F, c = 1..C, t1, t2 = 1..T, p = 1..P, r = 1..R$	(7C6)
	Delivery was completed in due time.	
CSP7	$Xp_{f,p} = \sum_{c=1}^C \sum_{t=1}^{TL} Xk_{f,c,p,t} \quad \forall f = 1..F, p = 1..P$	(1C7)
	Balance constraint 1-the volume of production in the factory $f$ is equal to the volume delivered to a distribution center $c$ .	
	$\sum_{f=1}^F \sum_{t=1}^T Xk_{f,c,p,t} = \sum_{r=1}^R \sum_{t=1}^T Yk_{c,r,p,t} \quad \forall c = 1..C, p = 1..P$	(2C7)
	Balance constraint 2-the volume delivered from distribution center $c$ is equal the volume delivered to customer $r$ .	
	$\sum_{p=1}^P \sum_{t=1}^T Xk_{f,c,p,t} = Xs_{c,p} \quad \forall c = 1..C, p = 1..P$	(3C7)
	Balance constraint 3- the volume delivered to distribution center $c$ is equal the volume handled by this distribution center.	
	$\sum_{c=1}^C \sum_{t=1}^T Yk_{c,r,p,t} = Zr_p \quad \forall r = 1..R, p = 1..P$	(4C7)
	Balance constraint 4 - the volume delivered to customer $r$ is equal the volume ordered by this customer.	
	$\sum_{y=1}^Y \sum_{t=1}^T Zk_{r,y,k,p} = Xr_{y,p} \quad \forall y = 1..Y, p = 1..P$	(5C7)
	Balance constraint 5- the volume delivered to recycling center $y$ is equal the volume handled in this center .	

**Table 7** Explicit forms of CSPs<sup>T</sup> with a description of the constraints

CSPs <sup>T</sup>	Constrains	
CSP <sup>T</sup>	$X_{f,p,c,r,t1,t2}^T = 0 \forall Xd_{f,p,c,r,t1,t2} = 0 \forall f = 1..F, p = 1..P, c = 1..C, r = 1..R, t1, t2 = 1..T$	7T
	$X_{f,p,c,r,t1,t2}^T \in C \forall f = 1..F, p = 1..P, c = 1..C, r = 1..R, t1, t2 = 1..T$	8T
	$Xb_{f,c,t}^T \in C \forall f = 1..F, c = 1..C, t = 1..T$	9T
	$Yb_{c,r,t}^T \in C \forall c = 1..C, r = 1..R, t = 1..T$	10T
	$Tc_c^T \in \{0, 1\} \forall c = 1..C$	11T
	Constraints provide binarity and integrality of decision variables.	
	$\sum_{c=1}^C \sum_{r=1}^R \sum_{t1=1}^T \sum_{t2=1}^T X_{f,p,c,r,t1,t2}^T \leq W_{f,p} \quad \forall f = 1..F, p = 1..P$	2Ta
	The production capacity of the factory $f$ is not exceeded.	
	$\sum_{p=1}^P Wa_{f,p} \cdot \left( \sum_{t1=1}^T \sum_{t2=1}^T \sum_{c=1}^C X_{f,p,c,r,t1,t2}^T \cdot Uz_{c,p} \right) \leq 1 \quad \forall f = 1..F$	2Tb
	The joint production capacity in the factory $f$ for all item $p$ .	
	$Z \sum_{f=1}^F \sum_{c=1}^C \sum_{t1=1}^T \sum_{t2=1}^T X_{f,p,c,r,t1,t2}^T = Z_{j,k} \quad \forall r = 1..R, p = 1..P$	3T
	Constraint enforces the realization of all customer orders.	
	$\sum_{f=1}^F \sum_{c=1}^C \sum_{t1=1}^T \sum_{t2=1}^T X_{f,p,c,r,t1,t2}^T = Z_{j,k} \quad \forall r = 1..R, p = 1..P$	21Ta
	The utilization capacity of the recycling center is not exceeded.	
	$\sum_{p=1}^P Wra_{y,p} \cdot \left( \sum_{r=1}^R \sum_{t=1}^T ZR_{r,y,p,t}^T \right) \leq 1 \quad \forall y = 1..Y$	21Tb
	The joint utilization capacity in the recycling centeryfor all item $p$ .	
	$\sum_{y=1}^Y \sum_{t=1}^T (ZR_{r,y,p,t}^T \cdot Ud_{r,p}) = Zr_p \quad \forall r = 1..R, p = 1..P$	21Tc
	A specified percentage of manufactured/sold items/products are recycled	
	$\sum_{f=1}^F \sum_{p=1}^P \sum_{r=1}^R \sum_{t1=1}^T \sum_{t2=1}^T X_{f,p,c,r,t1,t2}^T \leq V_c \cdot Tc_c^T \text{ for } c = 1..C$	4T

**Table 7** (continued)

	The distribution capacities for particular distribution centers $c$ are not exceeded.	
	$\sum_{p=1}^P (P_p \cdot \sum_{r=1}^R \sum_{t=1}^T ZR_{r,y,p,t}^T) \leq Tb_y^T \cdot Vb_y \quad \forall y = 1..Y$	23T
	The recycling capacities for particular recycling centers $y$ are not exceeded.	
	$\sum_{f=1}^F \sum_{c=1}^C Xb_{f,c,t}^T + \sum_{c=1}^C \sum_{r=1}^R Yb_{c,r,t}^T + \sum_{r=1}^R \sum_{y=1}^Y Zb_{r,y,t}^T \leq Zt_t \quad \forall t = 1..T$	6T
	During transportation cannot be used more means of transport than is available.	
	$Zb_{r,y,t} \leq Cw \cdot Za_{r,y,t} \quad \forall r = 1..R, y = 1..Y, t = 1..T$	22T
	The deliveries from the customer to the recycling center is carried out by the selected routes.	
	$\sum_{p=1}^P \sum_{r=1}^R \sum_{t=1}^T (P_t \cdot X_{f,p,c,r,t,t_2}^T) \leq Pt_{t_1} \cdot Xb_{f,c,t_1}^T \quad \forall f = 1..F, c = 1..C, t_2 = 1..T$	5Ta
	The right amount of means of transport to delivery items $p$ from the factory $f$ to the distribution center $c$ .	
	$\sum_{f=1}^F \sum_{p=1}^P \sum_{t_2=1}^T (P_p \cdot X_{f,p,c,r,t_1,t_2}^T) \leq Pt_{t_2} \cdot Yb_{c,r,t_2}^T \quad \forall c = 1..C, r = 1..R, t_2 = 1..T$	5Tb
	Constraint sets the right amount of means of transport $t$ for delivery items $p$ from the distribution center $c$ to the customer $r$ .	
	$Zb_{r,y,t}^T \cdot Pt_t \geq ZR_{r,y,p,t}^T \cdot P_p \quad \forall r = 1..R, y = 1..Y, p = 1..P, t = 1..T$	24T
CSP8 <sup>T</sup>	Constraint sets the right amount of means of transport $t$ for delivery items $p$ from the customer $r$ to the recycling center $y$ .	
	Exclusion C( $X_{f,p_1,c,r,t_1,t_2}, X_{f,p_2,c,r,t_1,t_2}, c$ ) $\forall p_1 \neq p_2, p_1, p_2 = 1..P, f = 1..F, c = 1..C, r = 1..R, t_1, t_2 = 1..T$	26T
	Constraint enables the distribution of exclusively one of the two selected items $p$ in the distribution center $c$ .	
	Exclusion F( $X_{f,p_1,c,r,t_1,t_2}, X_{f,p_2,c,r,t_1,t_2}, f$ ) $\forall p_1 \neq p_2, p_1, p_2 = 1..P, f = 1..F, c = 1..C, r = 1..R, t_1, t_2 = 1..T$	27T
	Constraint enables the production of exclusively one of the two selected items $p$ in the factory $f$ .	
	Exclusion Y( $X_{f,p_1,c,r,t_1,t_2}, X_{f,p_2,c,r,t_1,t_2}, y$ ) $\forall p_1 \neq p_2, p_1, p_2 = 1..P, f = 1..F, c = 1..C, r = 1..R, t_1, t_2 = 1..T$	28T
	Constraint enables the recycling of exclusively one of the two selected items in the recycling center $y$ .	
CSP9 <sup>T</sup>	$Po = \sum_{p=1}^P \sum_{r=1}^R (Z_{r,p} \cdot VP_p)$	12T
	Constraint determines a total volume of carrying products.	
	$Pm_r = \sum_{p=1}^P (Z_{r,p} \cdot P_p) \quad \forall r = 1..R$	13T
	Constraint determines the overall volume of products that must be delivered to each customer	
	$\sum_{f=1}^F \sum_{c=1}^C \sum_{t=1}^T (Pt_t \cdot Xb_{f,c,t}^T) \geq Po$	14T
	Capacity constraint determines the means of transport reserved for the transport / delivery from manufacturers to distributors, which must be greater than total demand.	
	$\sum_{c=1}^C \sum_{t=1}^T (Pt_t \cdot Yb_{c,r,t}^T) \geq Pm_r \quad \forall r = 1..R$	15T
	Capacity constraint determines the means of transport reserved for the transport / delivery from distributors to customer, which must be greater than demand for the customer.	
	$\sum_{f=1}^F \sum_{c=1}^C Xb_{f,c,t}^T + \sum_{c=1}^C \sum_{r=1}^R Yb_{c,r,t}^T + \sum_{r=1}^R \sum_{y=1}^Y Zb_{r,y,t}^T \geq R \min_t \quad \forall t = 1..T$	16T
	Constraint taking into account the lower bound of the required number of means of transport.	
	$\sum_{f=1}^F \sum_{c=1}^C Xb_{f,c,t}^T + \sum_{c=1}^C \sum_{r=1}^R Yb_{c,r,t}^T + \sum_{r=1}^R \sum_{y=1}^Y Zb_{r,y,t}^T \leq R \max_t \quad \forall t = 1..T$	17T
	Constraint taking into account the upper bound of the required number of means of transport.	
	$\sum_{f=1}^F \sum_{c=1}^C \sum_{t=1}^T Xb_{f,c,t}^T \geq FC$	18T
	Constraint estimates the minimum number of courses from factory $f$ to distribution center $c$	
	$\sum_{c=1}^C \sum_{r=1}^R \sum_{t=1}^T Yb_{c,r,t}^T \geq CM$	19T
	Constraint estimates the minimum number of courses from distribution center $c$ to customer $r$	
	$\sum_{f=1}^F \sum_{c=1}^C \sum_{t=1}^T Xb_{f,c,t}^T + \sum_{c=1}^C \sum_{r=1}^R \sum_{t=1}^T Yb_{c,r,t}^T + \sum_{r=1}^R \sum_{y=1}^Y \sum_{t=1}^T Zb_{r,y,t}^T \geq Srod$	20T
	Constraint estimates the minimum number of courses.	

**Table 8** The transformation of decision variables during the transition from CSPs to CSPs<sup>T</sup>

CSPs	CSPs <sup>T</sup>	Description
X <sub>pf, p, Xry, p, Xsc, p</sub>	X <sub>T<sub>f,p,c,r,t1,t2</sub></sub>	transformed decision variable is generated only for the technologically possible combinations of indexes.
X <sub>k<sub>f,c,p,t</sub></sub>	–	unnecessary
X <sub>b<sub>f,c,t</sub></sub>	X <sub>b<sub>f,c,t</sub></sub> <sup>T</sup>	the same meaning
Y <sub>k<sub>c,r,p,t</sub></sub>	–	unnecessary
X <sub>a<sub>f,c,p,t</sub></sub>	–	unnecessary
Y <sub>a<sub>c,r,p,t</sub></sub>	–	unnecessary
Y <sub>b<sub>c,r,t</sub></sub>	Y <sub>b<sub>c,r,t</sub></sub> <sup>T</sup>	the same meaning
T <sub>c</sub>	T <sub>c</sub> <sup>T</sup>	the same meaning
T <sub>rcy</sub>	T <sub>y</sub> <sup>T</sup>	the same meaning
Z <sub>k<sub>r,y,p,t</sub></sub>	Z <sub>r,y,p,t</sub> <sup>T</sup>	the same meaning
Z <sub>a<sub>r,y,t</sub></sub>	Z <sub>r,y,t</sub> <sup>T</sup>	the same meaning
Z <sub>b<sub>r,y,t</sub></sub>	Z <sub>r,y,t</sub> <sup>T</sup>	the same meaning

**Table 9** The transformation of constraints during the transition from CSPs to CSPs<sup>T</sup>

CSPs	CSPs <sup>T</sup>	Description
1C0	–	unnecessary
2C0	7T,8T,9T,10T,11T	the same meaning
1C1	2Ta	the same meaning
2C1	2Tb	the same meaning
3C1	3T	the same meaning
1C2	21Ta	the same meaning
2C2	21Tb	the same meaning
3C2	21Tc	the same meaning
1C3, 2C3, 3C3, 4C3	4T	the same meaning
1C4, 2C4, 3C4, 4C4	23T	the same meaning
1C5, 2C5, 3C5, 4C5	6T	the same meaning
1C6, 2C6, 3C6, 4C6	–	unnecessary- no need to bind decision variables.
5C6, 6C6, 7C6, 1C7	–	unnecessary- no need to bind decision variables.
2C7, 3C7, 4C7, 5C7	–	unnecessary- no need to bind decision variables.

## Appendix E

**Table 10** The results of numerical experiments for Fc=F1 and implementation A

Implementation A (multi-level approach)											
V <sub>1</sub> =1000, V <sub>2</sub> =1000, V <sub>3</sub> =2000, d <sub>1</sub> =70, d <sub>2</sub> =40, d <sub>3</sub> =30, d <sub>4</sub> =15											
N	F1 <sub>c</sub>	T	V(V <sub>int</sub> )	C	Vx <sub>1</sub>	Vx <sub>2</sub>	Vx <sub>3</sub>	dx <sub>1</sub>	dx <sub>2</sub>	dx <sub>3</sub>	dx <sub>4</sub>
90	143 426	285	3692(2433)	889	950	989	1991	68	40	30	15
75	103 533	173	3346(2087)	889	975	969	1331	43	33	22	15
60	71 918	143	2998(1739)	889	842	999	779	36	27	15	15
40	40 170	130	2540(1281)	889	640	1000	0	28	12	7	12
30	33 300	205	2304(1045)	889	410	900	0	18	15	9	7
20	22 093	16	2076(817)	889	0	780	0	14	10	6	4
15	18 082	8	1960(701)	889	0	655	0	14	9	5	3
10	10 128	6	1842(593)	889	0	330	0	16	4	1	2
5	6 270	4	1712(473)	889	0	125	0	9	1	0	1

**Table 11** The results of numerical experiments for Fc=F1 and implementation B

Implementation B (mathematical programming)				
$V_1 = 1000, V_2 = 1000, V_3 = 2000, d_1 = 70, d_2 = 40, d_3 = 30, d_4 = 15$				
N	F1 <sub>c</sub>	T	V(V <sub>int</sub> )	C
90	NFSF	600*	7992(6306)	30 549
75	184 346**	600*	7992(6306)	26 229
60	78 651**	600*	7992(6306)	21 909
40	40 445**	600*	7992(6306)	16 149
30	33 499**	600*	7992(6306)	13 269
20	22 107**	600*	7992(6306)	10389
15	18 082**	600*	7992(6306)	8 949
10	10 128	143	7992(6306)	7 509
5	6 270	10 s	7992(6306)	6 069

**Table 12** The results of numerical experiments for Fc=F2 and implementation A

Implementation A (multi-level approach)											
$V_1 = 1000, V_2 = 1000, V_3 = 2000, d_1 = 70, d_2 = 40, d_3 = 30, d_4 = 15$											
N	F2 <sub>c</sub>	T	V(V <sub>int</sub> )	C	Vx <sub>1</sub>	Vx <sub>2</sub>	Vx <sub>3</sub>	dx <sub>1</sub>	dx <sub>2</sub>	dx <sub>3</sub>	dx <sub>4</sub>
90	20 515	285	3692(2433)	889	1000	975	1955	67	40	30	15
75	17 345	318	3346(2087)	889	290	994	1991	50	30	28	15
60	13 797	268	2998(1739)	889	979	645	996	12	40	20	15
40	8 552	214	2540(1281)	889	798	87	755	6	28	4	14
30	7 546	148	2304(1045)	889	760	440	110	14	21	6	8
20	5 237	35	2076(817)	889	465	0	315	11	12	3	6
15	4 557	21	1960(701)	889	655	0	0	11	11	3	4
10	3 188	10	1842(593)	889	315	15	0	16	4	1	2
5	1 170	4	1712(473)	889	0	125	0	9	1	0	1

**Table 13** The results of numerical experiments for Fc=F2 and implementation B

Implementation B (mathematical programming)				
$V_1 = 1000, V_2 = 1000, V_3 = 2000, d_1 = 70, d_2 = 40, d_3 = 30, d_4 = 15$				
N	F2 <sub>c</sub>	T	V(V <sub>int</sub> )	C
90	31 456**	600*	7992(6306)	30 549
75	20 546**	600*	7992(6306)	26 229
60	14 860**	600*	7992(6306)	21 909
40	8 988**	600*	7992(6306)	16 149
30	8 149**	600*	7992(6306)	13 269
20	5 643**	600*	7992(6306)	10389
15	4 557	514	7992(6306)	8 949
10	3 188	204	7992(6306)	7 509
5	1 170	9 s	7992(6306)	6 069

**Table 14** The results of numerical experiments for Fc=F1 and implementation C

Implementation C (hybrid approach)				
$V_1 = 1000, V_2 = 1000, V_3 = 2000, d_1 = 70, d_2 = 40, d_3 = 30, d_4 = 15$				
N	F1 <sub>c</sub>	T	V(V <sub>int</sub> )	C
90	195 345**	600	6156(3156)	5123
75	174 216**	600	5878(2945)	5123
60	74 545**	600	5734(2846)	4956
40	40170	523	5556(2645)	4956
30	33 300	356	5445(2453)	4956
20	22 093	134	4745(2137)	4956
15	18 082	67	4389(1945)	4574
10	10 128	34	3967(1734)	4574
5	6 270	23	3472(1326)	4565

**Table 15** The results of numerical experiments for Fc=F2 and implementation C

Implementation C (hybrid approach)				
$V_1 = 1000, V_2 = 1000, V_3 = 2000, d_1 = 70, d_2 = 40, d_3 = 30, d_4 = 15$				
N	F2 <sub>c</sub>	T	V(V <sub>int</sub> )	C
90	21 456**	600*	6156(3156)	5123
75	18 521**	600*	5878(2945)	5123
60	13 797**	600*	5734(2846)	4956
40	8 552	568	5556(2645)	4956
30	7 546	456	5445(2453)	4956
20	5 237	234	4745(2137)	4956
15	4 557	89	4389(1945)	4574
10	3 188	45	3967(1734)	4574
5	1 170	25	3472(1326)	4565

## References

- Apt K (2003) Principles of constraint programming, Cambridge University Press, Cambridge
- Barth P, Bockmayr A (1998) Modelling discrete optimisation problems in constraint logic programming. *Ann Oper Res* 81:467. <https://doi.org/10.1023/A:1018981813157>
- Benhamou F, Jussien N, O'Sullivan BA (2007) Trends in Constraint Programming, ISBN: 978-1-905209-97-Wiley-ISTE
- Bocewicz G, Banaszak Z (2013) Declarative approach to cyclic steady states space refinement: periodic processes scheduling. *Int J Adv Manuf Technol* 67(1-4):137–155. <https://doi.org/10.1007/s00170-013-4760-0>
- Bockmayr A, Kasper T (2004) Branch-and-infer, a framework for combining CP and IP. *Constraint Integer Programm Oper Res/Comput Sci Interfaces Ser* 27:59–87. [https://doi.org/10.1007/978-1-4419-8917-8\\_3](https://doi.org/10.1007/978-1-4419-8917-8_3)
- Brandenburg M, Rebs T (2015) Sustainable supply chain management: a modeling perspective. *Ann Oper Res* 229:213. <https://doi.org/10.1007/s10479-015-1853-1>
- Caricato P, Grieco A (2009) A DSS for production planning focused on customer service and technological aspects. *Robot Comput-Integr Manuf* 25:871–878. <https://doi.org/10.1016/j.rcim.2009.06.003>
- Eclipse (2015) Eclipse - The Eclipse Foundation open source community website, Accessed May 4, <https://www.eclipse.org>
- Escudero LF, Martello S, Strusevich V (2013) An overview of computational issues in combinatorial optimization. *Ann Oper Res* 207:1. <https://doi.org/10.1007/s10479-013-1400-x>
- Grzybowska K, Gajšek B (2016) Regional logistics information platform as a support for coordination of supply chain. In: Bajo J et al (eds) Highlights of Practical Applications of Scalable Multi-Agent Systems, The PAAMS Collection, pp 61–72. [https://doi.org/10.1007/978-3-319-39387-2\\_6](https://doi.org/10.1007/978-3-319-39387-2_6)
- Hooker JN (2002) Logic, optimization, and constraint programming. *J Comput* 14(4):295–321
- Liess O, Michelon P (2008) A constraint programming approach for the resource-constrained project scheduling problem. *Ann Oper Res* 157(1):25–36. <https://doi.org/10.1007/s10479-007-0188-y>
- Nielsen I, Dang Q, Nielsen P, Pawlewski P (2014) Scheduling of mobile robots with preemptive tasks. *Adv Intell Syst Comput* 290:19–27. [https://doi.org/10.1007/978-3-319-07593-8\\_3](https://doi.org/10.1007/978-3-319-07593-8_3)
- Milano M, Wallace M (2010) Integrating operation research in constraint programming. *Ann Oper Res* 175(1):37–76.
- Roch J, Ramos C (1994) Task planning for flexible and agile manufacturing systems. *IROS '94* 1:105–112. <https://doi.org/10.1109/IROS.1994.407403>
- Rossi F, Van Beek P, Walsh T (2006) Handbook of constraint programming. Elsevier Science Inc, New York

17. Schrijver A (1998) Theory of linear and integer programming, ISBN 0-471-98232-6. Wiley, NY
18. Sitek P (2014) A Hybrid Approach to the Two-Echelon Capacitated Vehicle Routing Problem (2E-CVRP). In: Recent Advances In Automation, Robotics And Measuring Techniques, Advances in Intelligent Systems and Computing, vol 267, pp 251–263. [https://doi.org/10.1007/978-3-319-05353-0\\_25](https://doi.org/10.1007/978-3-319-05353-0_25)
19. Sitek P (2015) A hybrid approach to sustainable supply chain optimization. In: Progress in Automation, Robotics and Measuring Techniques, pp 243–254. [https://doi.org/10.1007/978-3-319-15796-2\\_25](https://doi.org/10.1007/978-3-319-15796-2_25)
20. Sitek P, Wikarek J (2013) A hybrid approach to modeling and, optimization for supply chain management with multimodal transport. In: IEEE Conference: 18th International Conference on Methods and Models in Automation and Robotics (MMAR), pp 777–782. <https://doi.org/10.1109/MMAR.2013.6670011>
21. Sitek P, Wikarek J (2015) 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. <https://doi.org/10.1080/00207543.2015.1005762>
22. Sitek P, Wikarek J (2016) A hybrid programming framework for modeling and solving constraint satisfaction and optimization problems. In: Scientific Programming, Volume 2016, Article ID 5102616. <https://doi.org/10.1155/2016/5102616>
23. Sitek P, Nielsen IE, Wikarek J (2014) A hybrid multi-agent approach to the solving supply chain problems. Procedia Comput Sci KES 35:1557–1566. <https://doi.org/10.1016/j.procs.2014.08.239>
24. Teorey T, Lightstone S, Nadeau T, Jagadish H (2011) Database Modeling and Design: Logical Design, The Morgan Kaufmann Series in Data Management Systems 5th Edition, ISBN-13: 978-0123820204
25. Tsang E (1993) Foundation of constraint satisfaction. Academic Press, London and San Diego
26. Wikarek J (2016) An integrated declarative approach to decision support for scheduling groups of jobs. In: Information Systems Architecture And Technology, ISAT 2015, PT I Advances in Intelligent Systems and Computing, vol 429, pp 53–64. [https://doi.org/10.1007/978-3-319-28555-9\\_5](https://doi.org/10.1007/978-3-319-28555-9_5)



**Paweł Sitek** is an assistant professor at the Department of Electrical Engineering, Automatic Control and Computer Science of Kielce University of Technology in Poland. He obtained M.Sc degree in Electrical Engineering from Kielce University of Technology, Poland and a Ph.D. from Silesian University of Technology, Poland. His research interests focus on operation research, constraints programming techniques, production planning and scheduling, discrete optimization, and decision support systems. He is the co-author of the concept of hybrid approach to modeling and solving constraint problems. He is also an author and co-author over 130 manuscripts including international journals, chapters in books and conference proceedings. He is a reviewer for a number of international journals - *International Journal of Production Research*, *Decision Sciences*, *IEEE Transactions on Systems, Man and Cybernetics: Systems*, *Journal of Intelligent Manufacturing*, *Computers & Operation Research*, etc. He is a reviewer and a member of the program committees of numerous conferences.



**Jarosław Wikarek** is an assistant professor at the Department of Electrical Engineering, Automatic Control and Computer Science of Kielce University of Technology in Poland. He obtained M.Sc degree in Electrical Engineering from Kielce University of Technology, Poland and a Ph.D. from Silesian University of Technology, Poland. His research interests focus on operation research, constraints programming techniques, production planning and scheduling, manufacturing, ERP systems, and decision support systems. He is the co-author of the concept of hybrid approach to modeling and solving constraint problems. He is also an author and co-author over 100 manuscripts including international journals, chapters in books and conference proceedings.

ing, manufacturing, ERP systems, and decision support systems. He is the co-author of the concept of hybrid approach to modeling and solving constraint problems. He is also an author and co-author over 100 manuscripts including international journals, chapters in books and conference proceedings.