A hybrid approach to modeling and solving decision and optimization problems

Pawel Sitek, Kielce University of Technology

Kielce University of Technology, Poland

Kielce University of Technology, Poland

- Exists since 1965
- 4 500 students studying at five faculties
- Civil Engineering and Architecture
- Electrical Engineering, Automatic Control and Computer Science
- Environmental Engineering, Geomatics and Renewable Energy
- Mechatronics and Mechanical Engineering
- Management and Computer Modelling
- 450 teaching staff

- Modern learning and training facilities:
  Laser Processing Research Centre,
  High-tech laboratories, library,
  sports centre, etc.

- Active cooperation with over 100 foreign universities in Europe and around the whole world

Kielce University of Technology, Poland

- Since 2019, after the reform of higher education in Poland, the so-called Doctoral Schools have been established.

- A Doctoral School operates at KUT, which allows you to obtain a doctoral degree in the following disciplines:
- **Mechanical Engineering,**
- **Civil Engineering,**
- **Environmental Engineering,**
- **Automation, Electronics and Electrical Engineering.**

Decision problems

- Decision problems are common in everyday life and other fields, such as industry, finance, business, trade, education and transportation, to name only a few.

- Many decision problems can be formulated as feasibility or optimization problems, in which the question is to decide the values of various decision variables representing a quantity, time, allocation, selection, location, etc.

Distribution problems

- Distribution processes, defined as the flow of certain items: goods, services, packages, parcels, financial means, information, persons, etc., over a given distribution network, are common in nearly all areas of contemporary man's activity.

Distribution problems

- Generally, a distribution problem is defined as a transfer of items (goods, services, packages, parcels, data, information, persons, etc.) from the point of origin (factories, importers, processing plants, individual suppliers, servers, etc.) through intermediary points (distribution centers, depots, routers, docks, warehouses, sorting offices, etc.) to delivery points (stores, individual customers, computers, mobile devices, etc.).

- The supply is realized over a distribution network, which can be a road network, a rail network, a sea network, a computer network, energy/gas/water/sewage network infrastructure, and multimodal networks combining several individual network types.

Distribution problems

- The primary questions to be answered for the given distribution process (logistic) to be efficient are as follows:
- *What is the optimal cost of delivering products to customers?*
- *Is it possible to deliver goods to customers in given the set time and distribution network?*
- *What is the optimal distribution network for the given volume of deliveries?*
- *Is it possible to deliver goods or services to customers using incomplete distribution network? etc.*
- *What is the optimal distribution of stops in City's Transport Routes?*

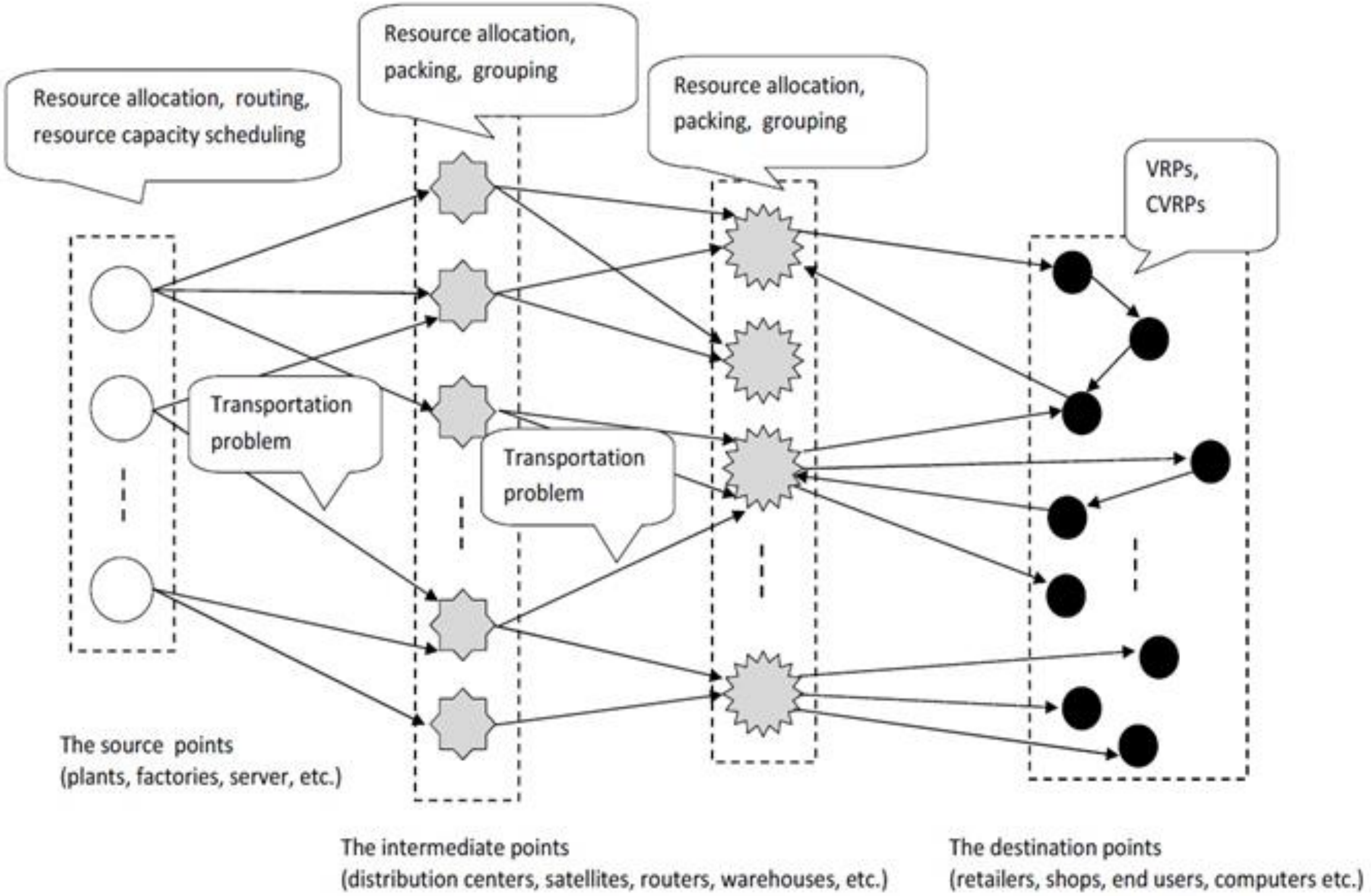
Distribution problems

- A distribution process very often consists of **multiple sub-processes**, such as **transportation, packaging and grouping, routing, scheduling and resource allocation**, etc. may contribute to the complexity of the distribution process at every stage or in each area, problem or process.

- Considered above problems are characterized by the following features:
  - **Discrete** and **deterministic** character;
  - **Large sizes** (i.e., with thousands of decision variables);
  - **Relational** or equivalent model of data structures;
  - **Many types of constraints** (equations and inequalities that are linear and integer, nonlinear, logical, and symbolic) that involve many decision variables;
  - **Demand exact solutions.**

Distribution problems

- A general and simplified diagram of a distribution process is shown in Fig. 1.



Mathematical programming

- In practice, most distribution problem models are mathematical programming (MP) models based on linear programming-based approach, in particular on **mixed integer linear programming (MILP)** or **integer linear programming (ILP)**.

Linear or Integer programming (one formulation)

find x to minimize $z = c^T x$ **cost** or **objective** function

subject to $Ax \leq b$ **inequalities**

$x \geq 0$

$c \in R^n, \quad b \in R^m, \quad A \in R^{n \times m}$

Linear programming:

$x \in R^n$ (polynomial time)

Integer programming:

$x \in Z^n$ (NP-complete)

Extremely general framework, especially IP

$$\min_{x} \begin{bmatrix} -160 & -50 & -50 & -70 & -110 \end{bmatrix} x$$

$$\begin{bmatrix} 8 & 3 & 3 & 3 & 6 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix} x \leq \begin{bmatrix} 10 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$x \in \mathbb{Z}^n$$

Mathematical programming

- Primary drawbacks of mathematical programming based distribution problem models (MILP, IP, ILP etc.) include linearity of constraints and limited efficiency of combinatorial optimization for problems of larger size (NP-complete).

Declarative programming

- **A declarative programming** is a programming paradigm—a style of building the structure and elements of computer programs—that expresses the logic of a computation without describing its control flow.
- Declarative paradigm-based languages describe **what** a program needs to accomplish in terms of a problem domain, rather than describing **how** to achieve it as a sequence of programming language primitives (which is left to the implementation of the language).
- This is in contrast to **imperative programming (C, Java, Basic, etc.)**, which implements algorithms in explicit steps.

Declarative programming

- Declarative programming languages often considers programs as theories of a formal logic.

- Common declarative languages include those of database query languages (e.g., SQL, XQuery), logic programming (e.g. Prolog, Datalog, answer set programming), etc.

- Declarative programming includes a number of better-known programming paradigms such **as Constraint programming, Functional programming**, etc.

Constraint programming

- Constraint programming/Constraint logic programming states relations between variables in the form of constraints that specify the properties of the target solution.

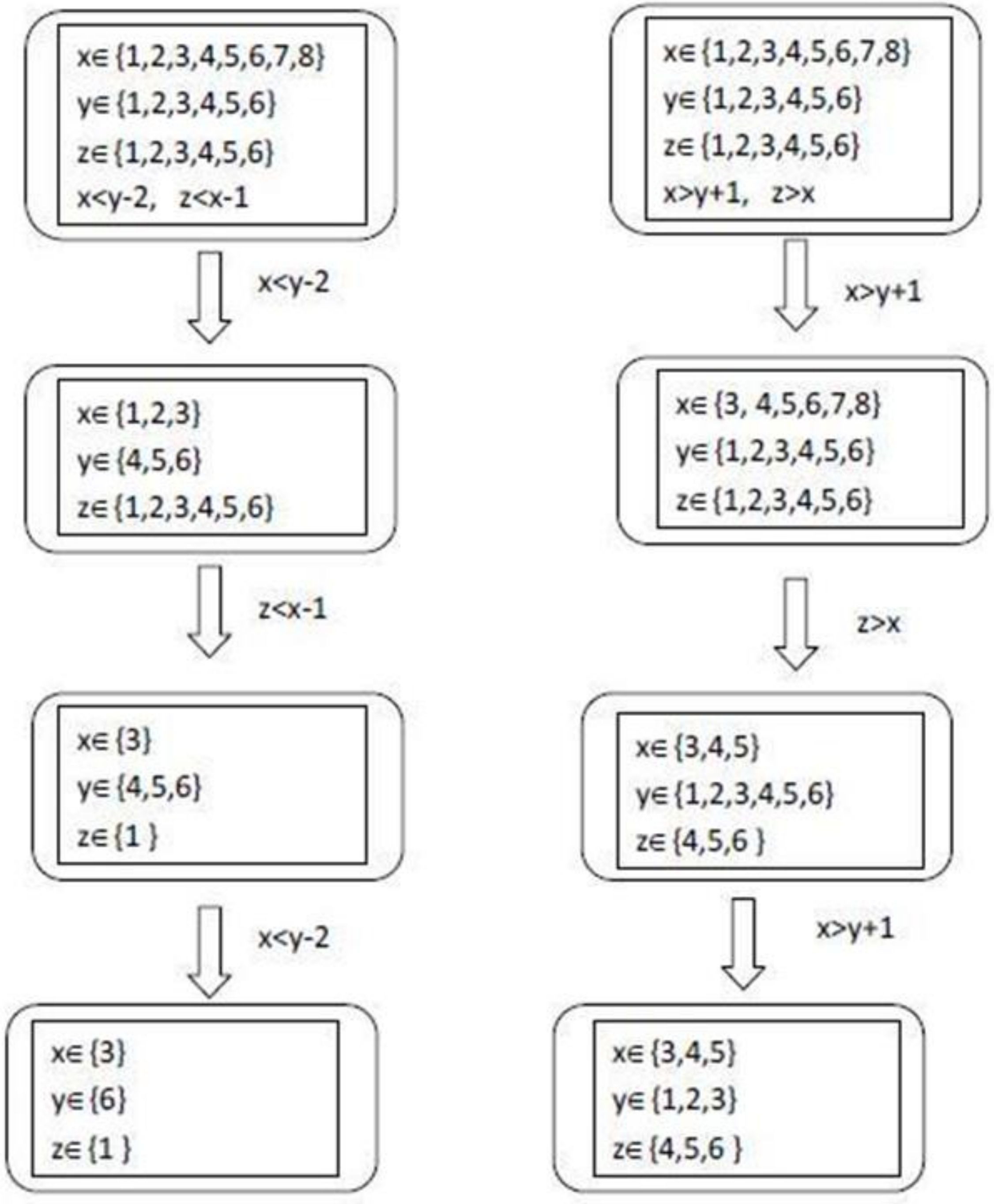
- The set of constraints is solved by giving a value to each variable so that the solution is consistent with the maximum number of constraints.

Constraint programming

- Declarative environments are free of many MP shortcomings.
- Declarative environments and especially CLP (Constraint Logic Programming) are very effective in modeling various types of constraints: linear, nonlinear, logical, etc.;
- CLP is also characterized by high efficiency in finding feasible solutions, i.e. those that answer general questions such as: *Is it possible to..?*
- Declarative environments show some limitations when numerous decision variables are added up in constraints. Such constraints are common in distribution.

Constraint programming

- In CLP are Inbuilt methods/techniques for removing infeasible values from variable domains (filtering, domain consistency maintenance, passes reduced domains to next constraint (**constraint propagation**);



Comparison of MP and CP/CLP - advantages and disadvantages

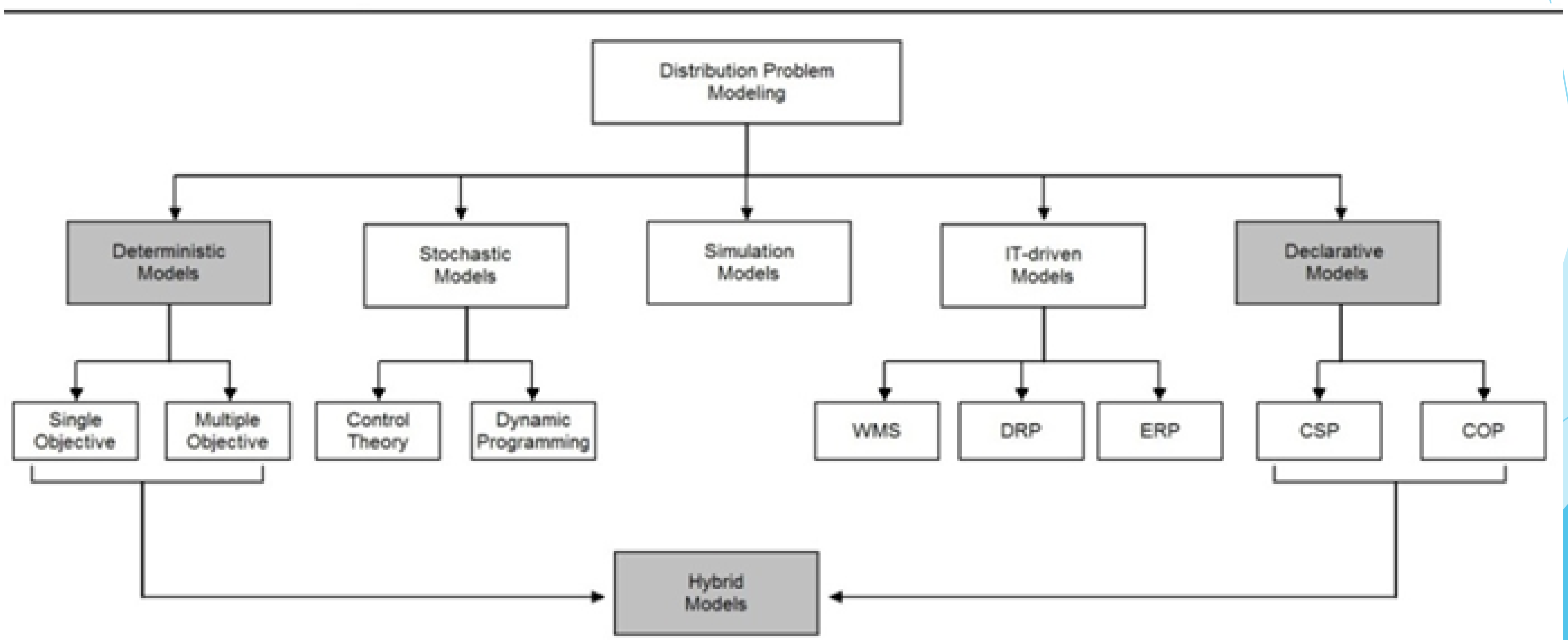
- Advantages and disadvantages of both environments in the context of modeling and solving distribution problems:

MP	CLP
Advantages	
Relaxation methods - very effective when constraints and/or objective function contain many variables. For example, cost and profit.	The more constraints, the better but with a specific structure, e.g. binary constraints.
Tools for filtering. Duality theory.	Modeling easy. More powerful modeling language. Any type of constraints, questions and so on.
MP's focus on special classes of problems is useful for solving relaxations or subproblems (vertical structure).	Simpler models (due to global constraints). Constraints convey problem structure to the solver. CP's idea of global constraint exploits structure within a problem (horizontal structure). Data-driven approach.
More robust and good with continuous variables. Numerical calculations.	Inbuilt methods for removing infeasible values from variable domains (filtering, domain consistency maintenance, passes reduced domains to next constraint (constraint propagation)).

Disadvantages	
Only linear constraints.	Lacks relaxation technology, which can be critical when solving large-scale problems.
Artificial methods of modeling constraints allocation by the introduction of 0-1 variables.	Weaker for continuous variables (due to lack of numerical techniques).
Not very high efficiency in solving combinatorial problems with large sizes.	May fail when constraints contain many variables. These constraints do not propagate well.
Model completely separate from data - versatility is its advantage, disadvantage being that the model is far from the problem.	Often not good for finding optimal solutions due to lack of relaxation technology.

Hybrid approach

- Analysis of both environments (MP and CLP) indicates certain complementarity between the paradigms of both environments and encourages their **integration or hybridization**.



Hybrid approach
-the concept and assumptions

- Considering:
- (a) the key characteristics of distribution related problems, i.e. high complexity, discrete character, a number of different constraint types, including those that relate many variables, etc.;
- (b) complementarity of CLP and MILP environments;
- (c) existing schemes and CLP/MILP integration implementations

the original **hybrid approach** is proposed.

Hybrid approach -the concept and assumptions

- The key feature of this hybrid approach is that the problem is represented in the **form of facts**.
- Models for distribution problems are typically written as algebraic expressions with coefficient matrices and vectors of decision variables, as in mathematical programming.
- The new representation (facts) reduces the size of the problem. This reduction can be referred to as “hidden” transformation resulting only from the change in problem representation.
- With algebraic notation, the status is usually shown as a matrix of coefficients and the decisions as a vector of decision variables. If a status is unacceptable, its corresponding parameter is “0”. Then the corresponding decision parameters are zeroed but still present in the model representation.

Hybrid approach
-the concept and assumptions

Algebraic representation

$$\begin{pmatrix} c11 & c12 & c13 & c14 \\ c21 & c22 & c23 & c24 \\ c31 & c32 & c33 & c34 \\ c41 & c42 & c43 & c44 \end{pmatrix} \Rightarrow \begin{pmatrix} 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \Rightarrow$$

16 parameters

Facts-based representation

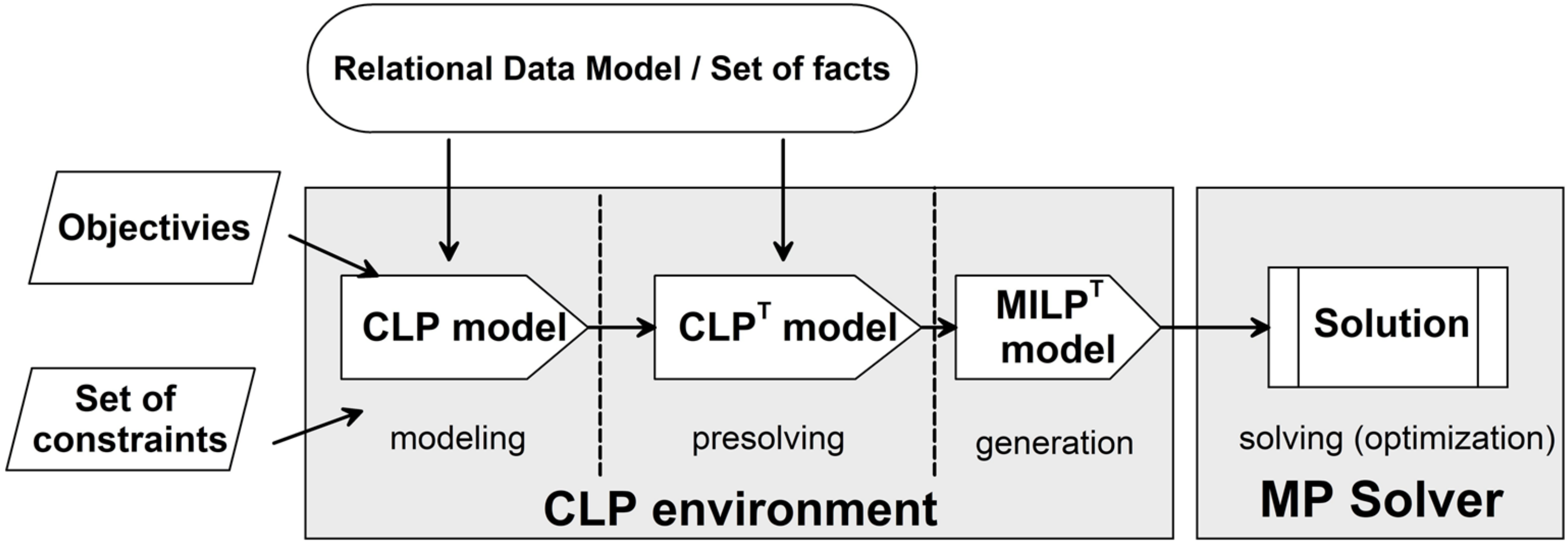
F_c(c_{12},1), F_c(c_{14},2), F_c(c_{23},1),F_c(c_{31},3), F_c(c_{32},2),F_c(c_{44},1) ⇒ 6 fact instances

Hybrid approach
-the concept and assumptions

- The main assumptions of the proposed hybrid approach are the following:
- Representation of the problem being modeled in the form of facts with a suitable structure following the relational model;
- Modeling problems using the values of fact attributes and relationships between attributes;
- Introduction of transformation methodology (using CLP techniques like constraint propagation, etc.), the key issue on the hybrid approach, which is a method of transformation and generation of models and presolving.

Hybrid implementation framework

- The hybrid framework comprises several phases: modeling, presolving, generating and solving. There are two inputs and the set of facts.
- Inputs are the objectives and the set of constraints to the modeled problem. Based on them, the original model of the problem is generated as a CLP model. Than CLP model is pre-solved.
- The built-in CLP method (constraint propagation) and the method of problem transformation are used for presolving.
- Presolving procedure results on the transformed model CLP^T.
- Transformed model is the basis for the automatic generation of the $MILP^T$ (Mixed Integer Linear Programming) model, which is solved in MP (Mathematical Programming) (with the use of an external solver or build in solver).
- MP Solvers: LINGO or GUROBI
- CLP : Eclipse CLP



Hybrid approach -computational examples

- The examples of distribution problems were selected for the hybrid approach test.
- The first is an SCM (Supply Chain Management) problem, which is currently one of the most important issues in modern logistics.
- The other is a 2E-CVRP (Two-Echelon Capacitated Vehicle Routing Problem), which is one of the more advanced variants of the VRP (Vehicle routing problem). VRPs are very common in logistics, production, transportation, etc.

Computational example

- The experiments were conducted for each model in two ways .
- The first one is directly applying the MP solver to the MILP model (a) and the second one is using the hybrid approach (b).
- The experiments were performed for the following illustrative examples: SCM (7 Factories, 3 Distributors, 10 Customers, 3 Means of Transport, between 10-100 Products) and 2E-CVRP (1 Depot, 2 Satellites, 12 Customers)

Computational example-SCM

- SCM problem is the management of the flow of goods and services, involves the movement and storage of raw materials, of work-in-process inventory, and of finished goods from point of origin (a) to point of consumption (c) via intermediate points (b) using transportation network.
- The objective function (Fc) defines the aggregate costs of the entire chain and consists of four elements (distributor, manufacturer, delivery)

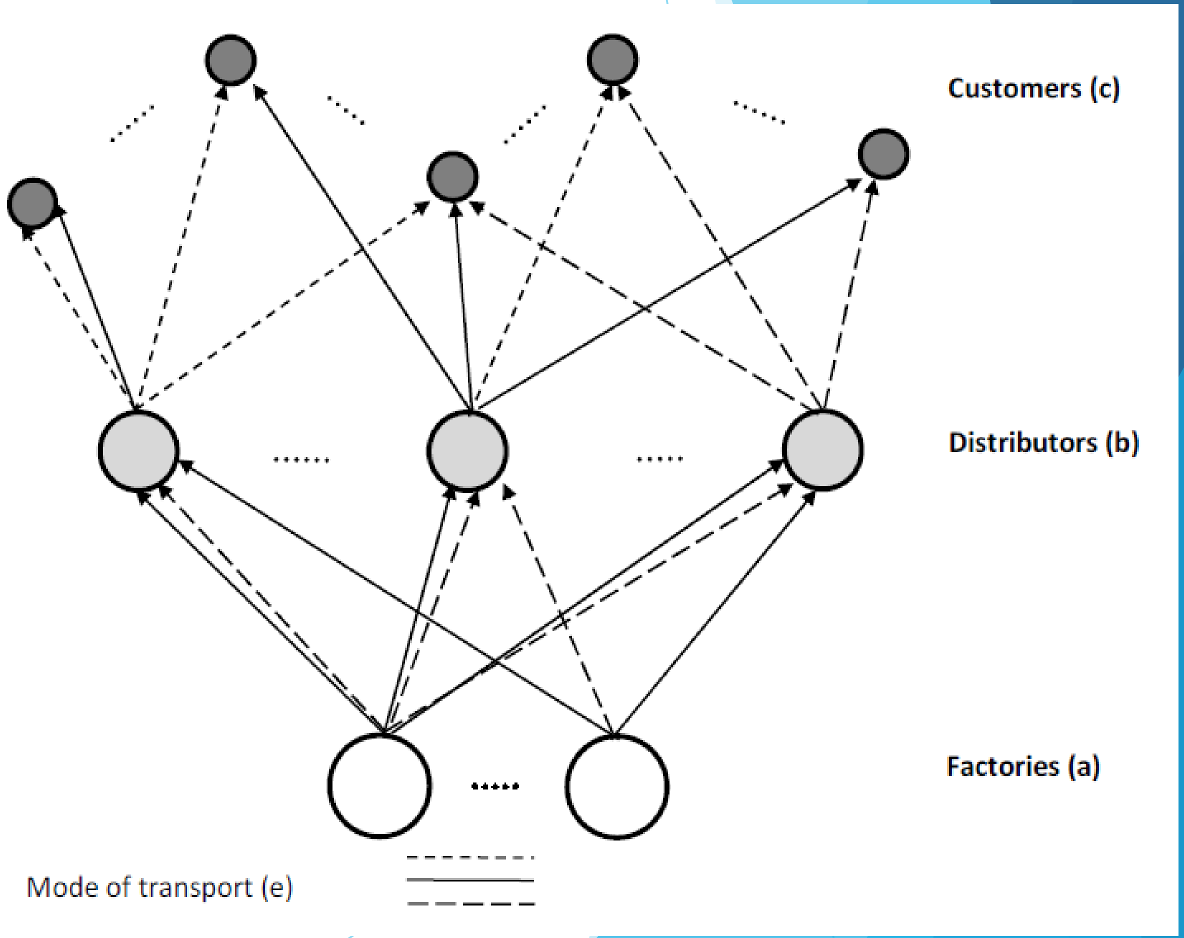
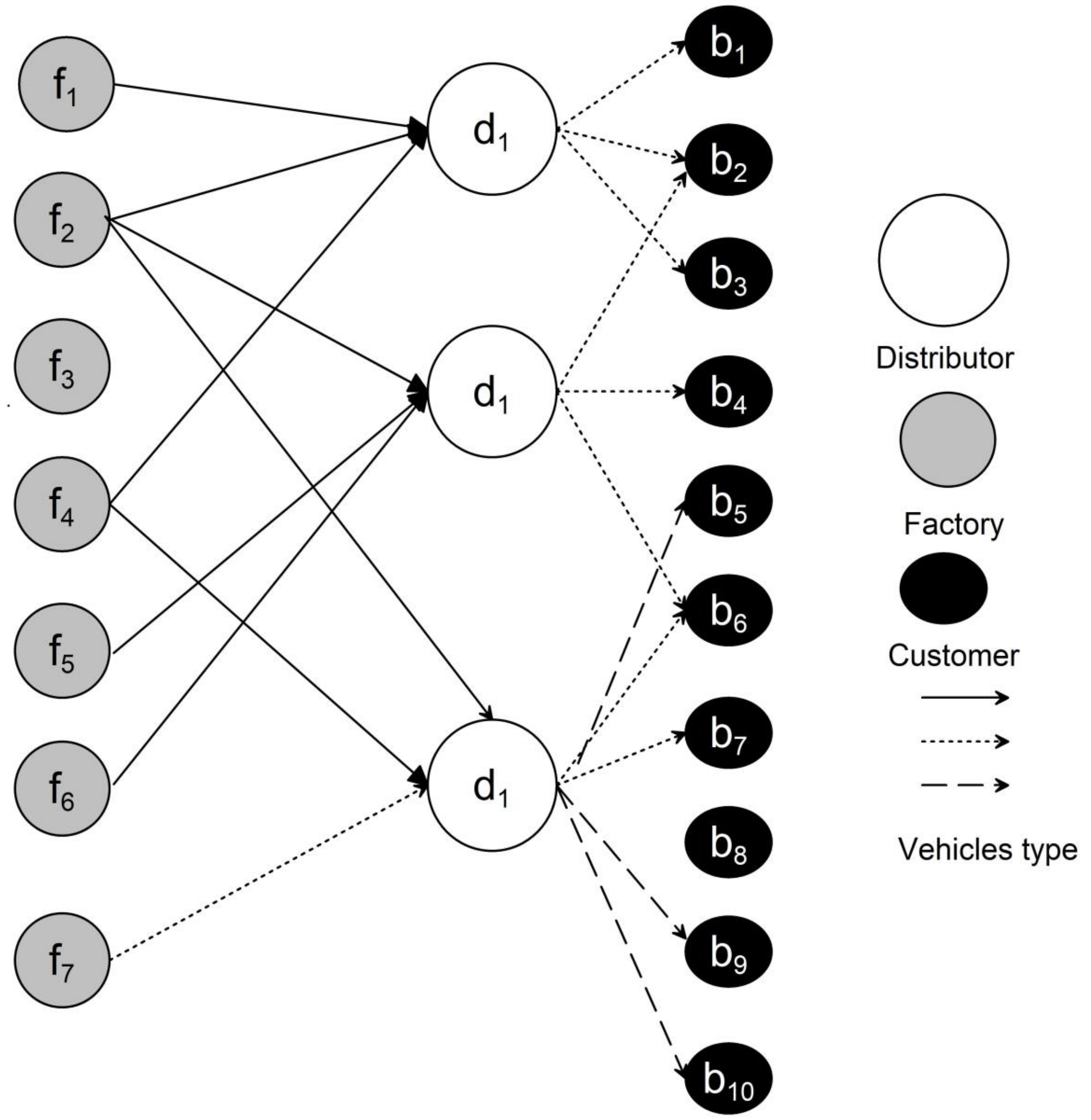


Fig. The basic supply chain network: (a)-factory, depot (b)-distribution center, ware-house, distributor (c)-customer, retailer

Computational example-SCM

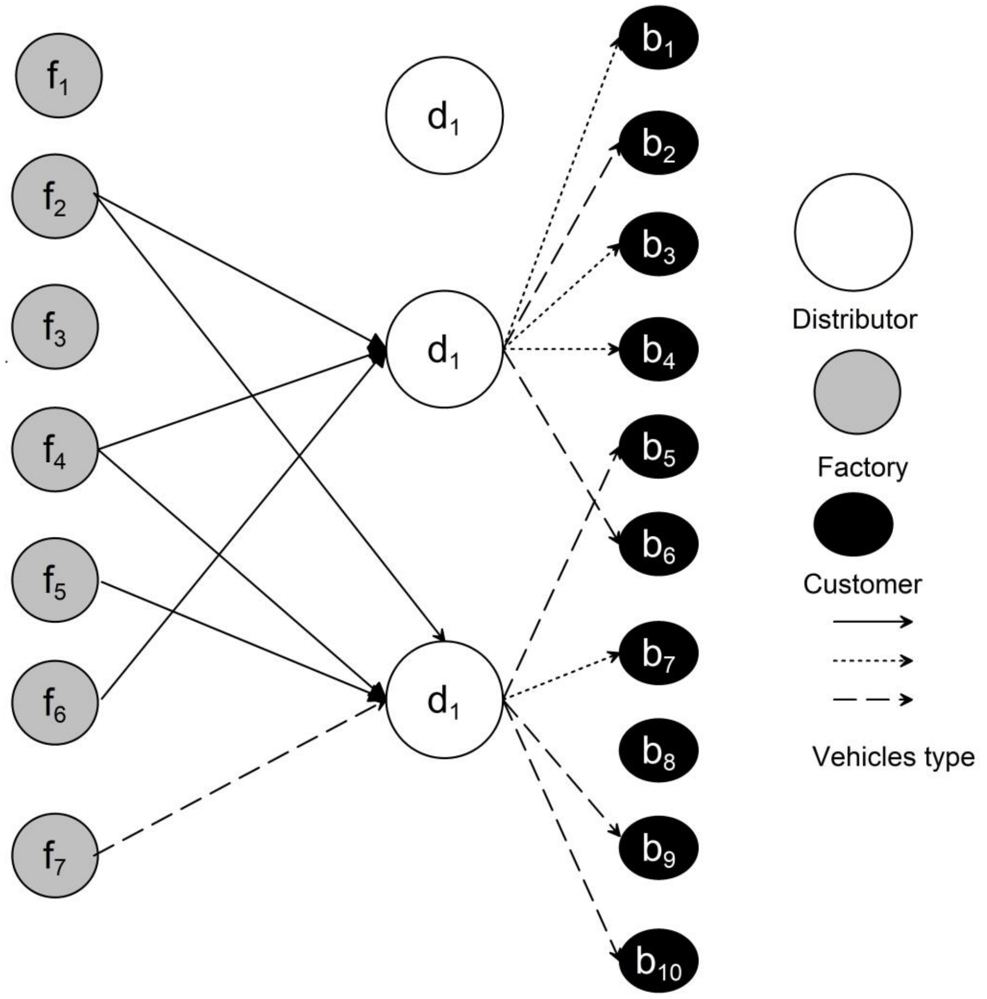
SCM (7 Factories, 3 Distributors, 10 Customers, 3 Means of Transport, between 10-100 Products)



Optimal solution

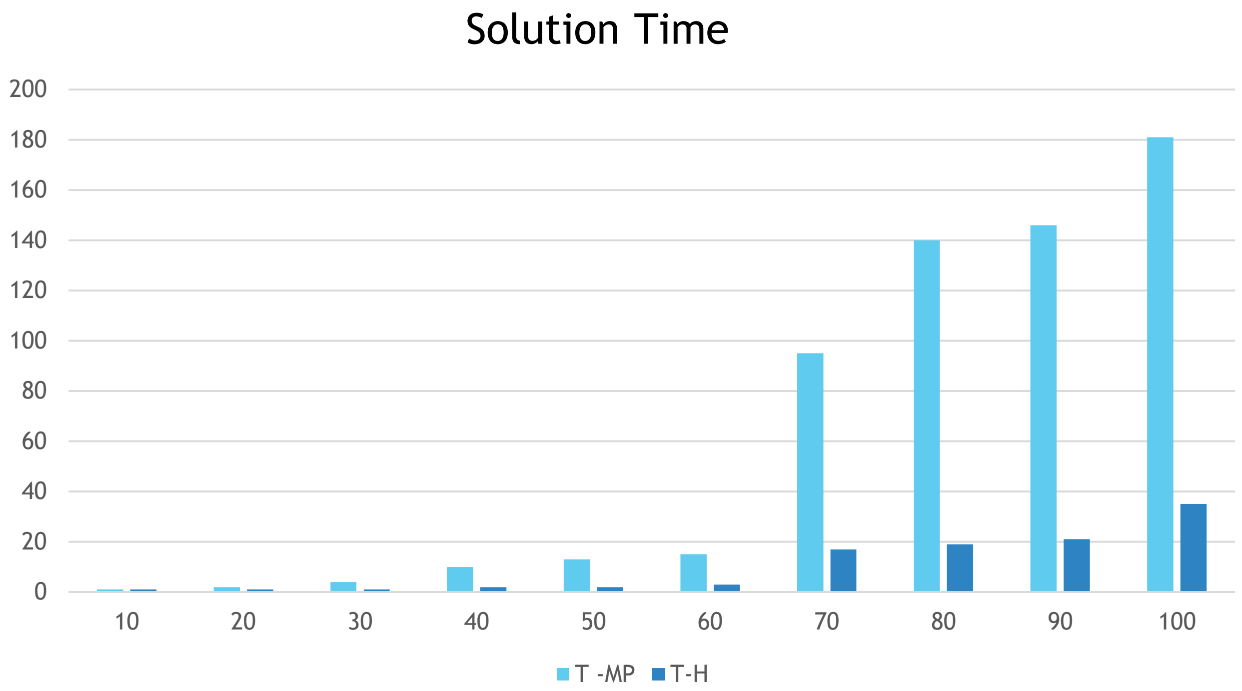
Computational example-SCM

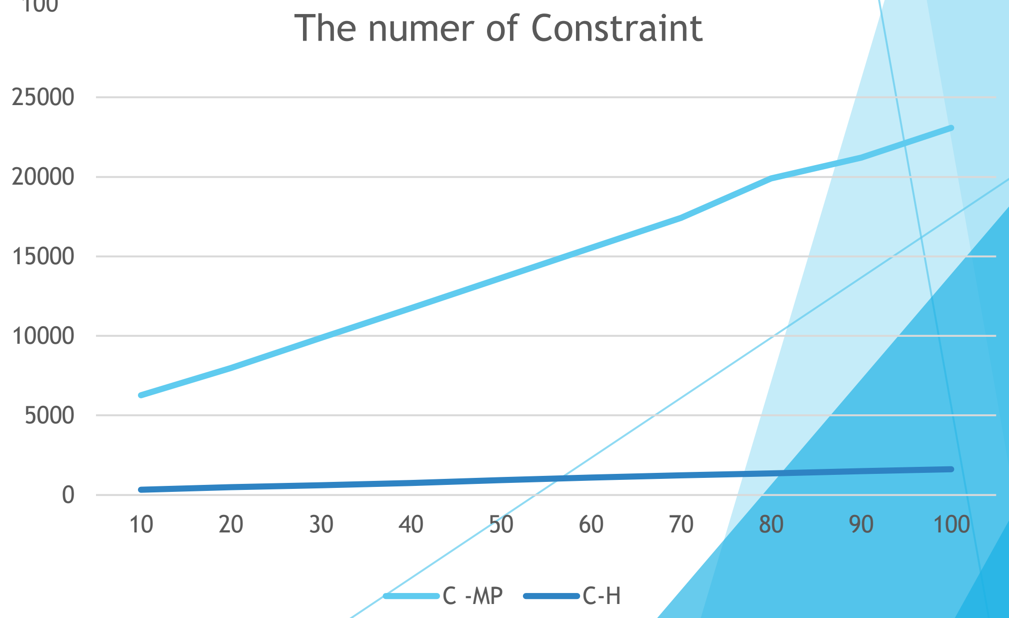
SCM (7 Factories, 3 Distributors, 10 Customers, 3 Means of Transport, between 10-100 Products)



Feasible solution

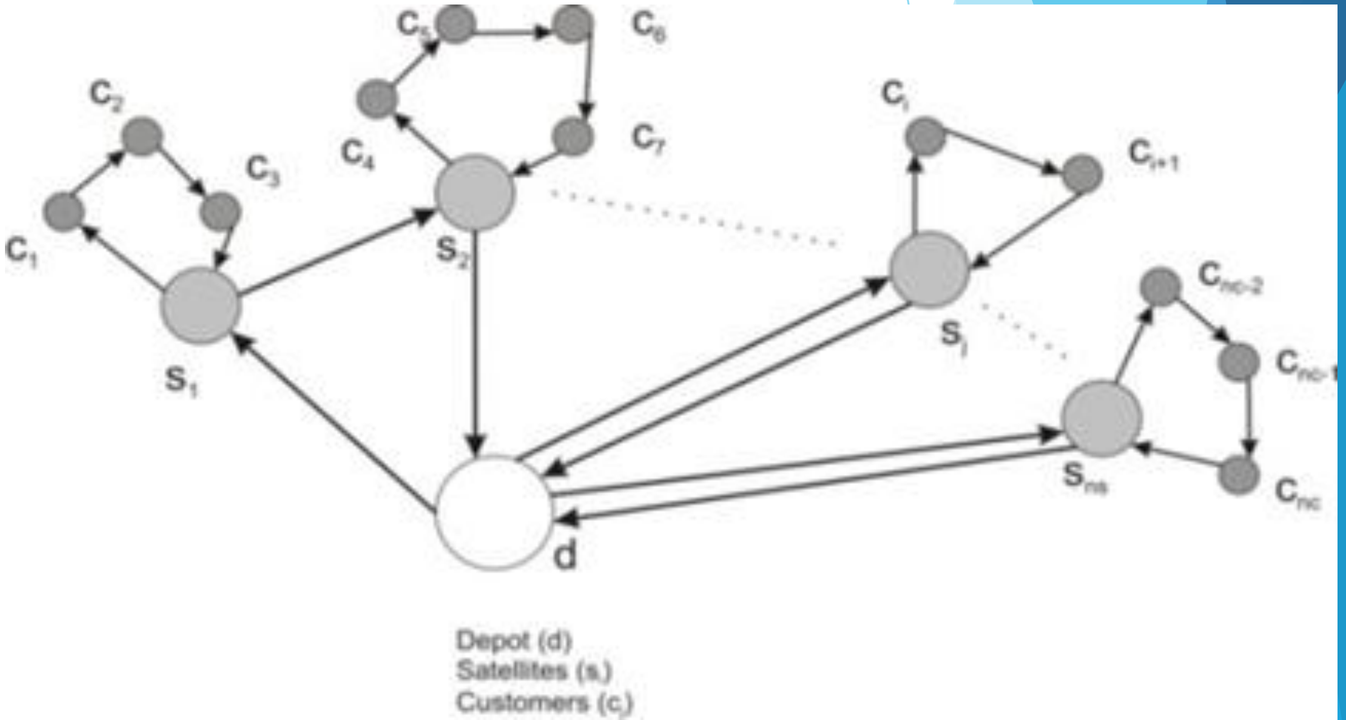
Computational example-SCM





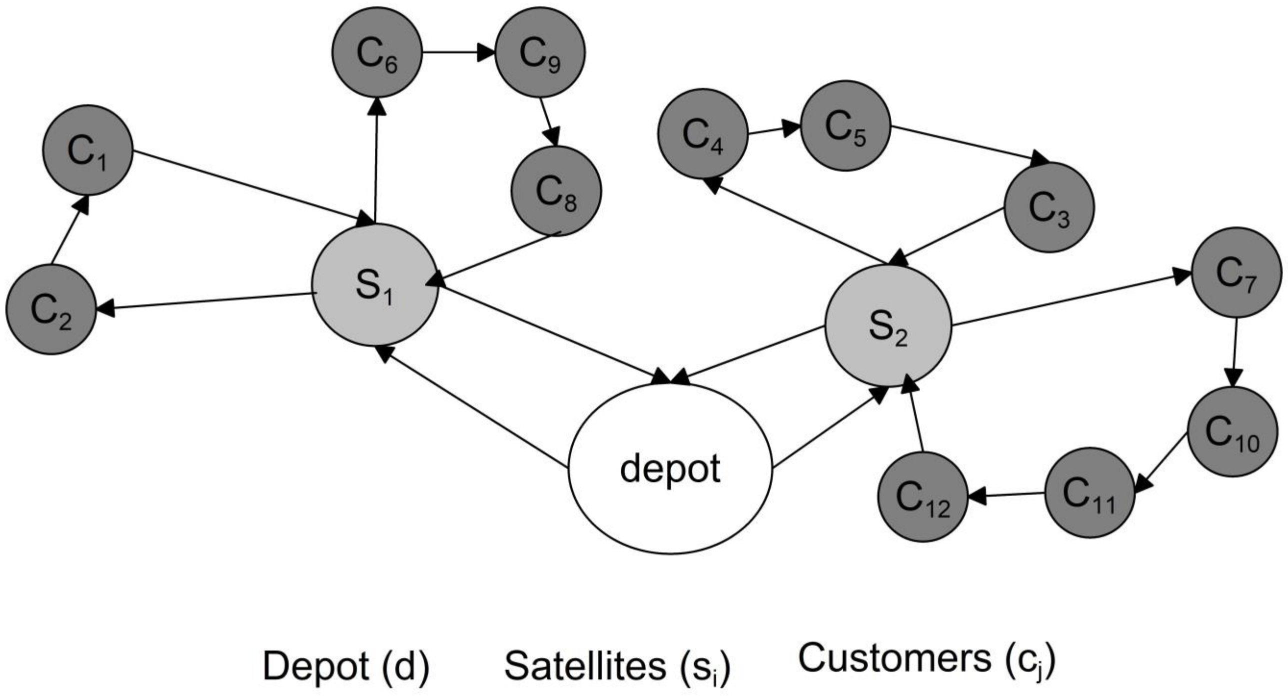
Computational example-2E-CVRP

- The Two-Echelon Capacitated Vehicle Routing Problem (2E-CVRP) is an extension of the classical Capacitated Vehicle Routing Problem (CVRP) where the delivery depot-customers passes through intermediate depots (called satellites).
- As in CVRP, the goal is to deliver goods to customers with known demands, minimizing the total delivery cost in the respect of vehicle capacity constraints.
- The transportation network is decomposed into two levels the 1st level connecting the depot (d) to intermediate depots (s) and the 2nd one connecting the intermediate depots (s) to the customers (c).
- The objective function (Fc) is to minimize the total transportation cost of the vehicles involved in both levels



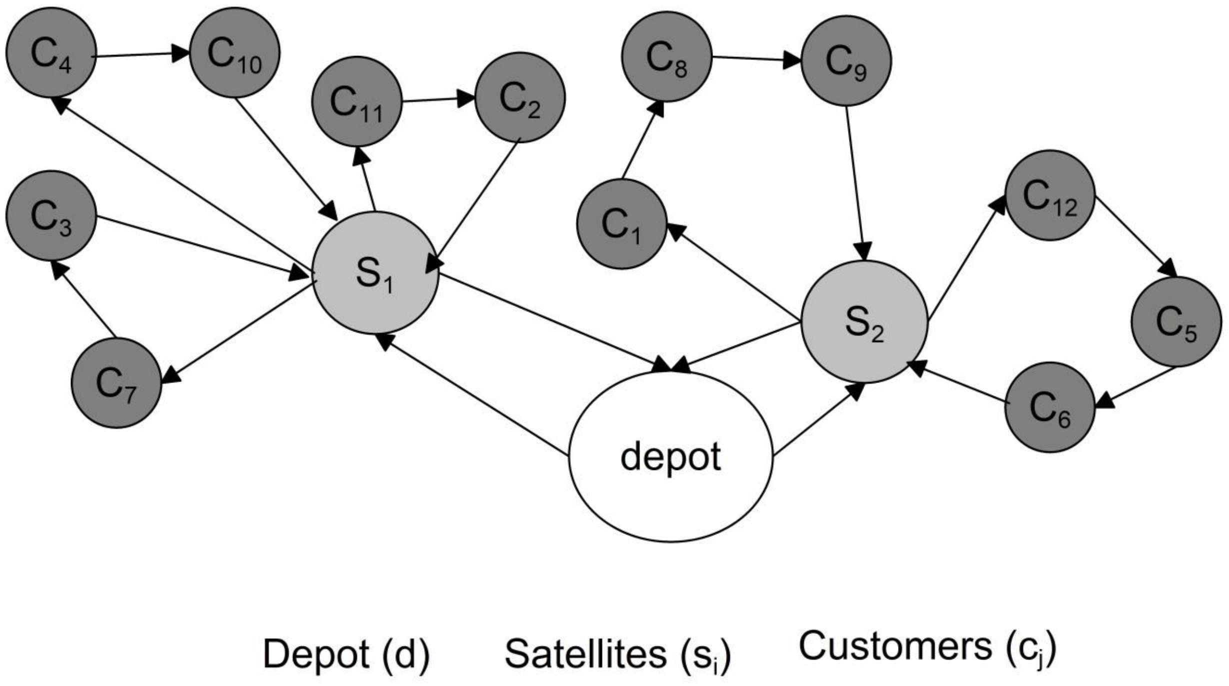
Computational example-2E-CVRP

2E-CVRP (1 Depot, 2 Satellites, 12 Customers)
Optimal solution Fc=288



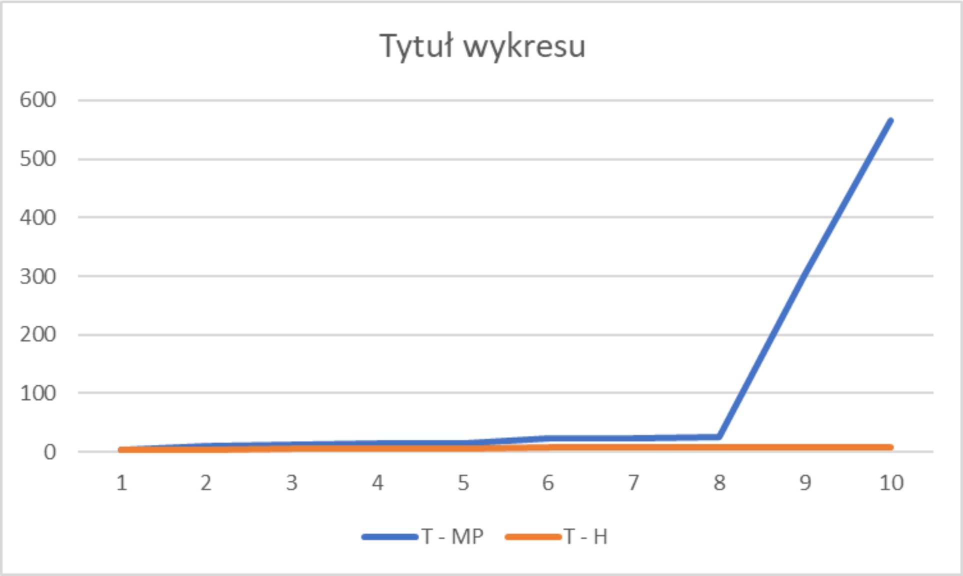
Computational example-2E-CVRP

2E-CVRP (1 Depot, 2 Satellites, 12 Customers)
Feasible solution Fc=334

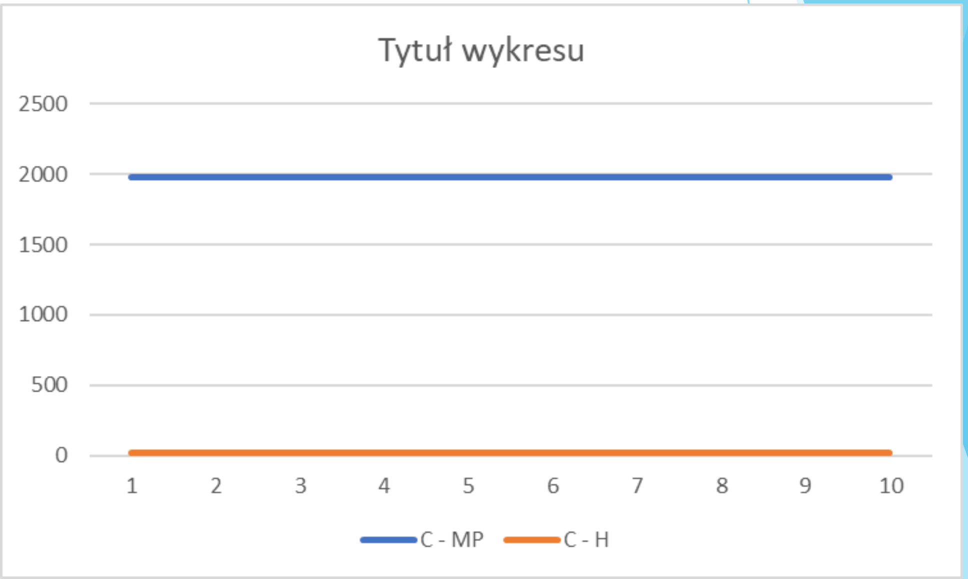


Computational example-2E-CVRP

Solution Time



The numer of constraints



Thank you for attention