CrossMark

Capacitated vehicle routing problem with pick-up and alternative delivery (CVRPPAD): model and implementation using hybrid approach

Pawel Sitek[1] · **Jarosław Wikarek**[1]

Abstract The paper presents an optimization model and its implementation using a hybrid approach for the Capacitated Vehicle Routing Problem with Pick-up and Alternative Delivery (CVRPPAD). The development of the CVRPPAD was motivated by postal items distribution issues. The model proposed combines various features of Vehicle Routing Problem variants. The novelty of this model lies in the introduction of alternativeness of item delivery points, differentiation of delivery point types in terms of their capacity (parcel lockers) and possibility of collecting the items during the execution of delivery routes. The model also takes into account a possibility of introducing time windows related to delivery time. The proposed data structure in the form of sets of facts facilitates the implementation of the model in all environments, Constraint Logic Programming (CLP), Mathematical Programming (MP), metaheuristics, databases, etc. The model is implemented in the MP, hybrid CLP/MP, and hybrid CLP/heuristic environments. The hybrid CLP/MP approach is the authors' original solution, which has already been used to solve Supply Chain Management problems, scheduling problems, routing problems etc. For large size problems, considering their combinatorial character, the proposed CLP/MP approach is ineffective. Its effectiveness will improve when MP is replaced by a heuristic. All implementations were performed on the same data instances (facts), which made it possible to compare them according to the solution search time, number of decision variables and number of constraints. The directions into which the model can be further developed were also presented.

Keywords VRP · CVRP · Constraint logic programming, mathematical programming · Hybrid approach · Optimization · Heuristics

✉ Pawel Sitek
  sitek@tu.kielce.pl

[1] Department of Information Systems, Kielce University of Technology, Kielce, Poland

⚡ Springer

1 Introduction

Observed in the last 10 years, fast development of information technologies, online services and mobile applications has resulted in a significant growth of e-commerce segment. The base of the European e-commerce customers who have ordered goods or services over the Internet for private use (2014) ranges from 15% in Bulgaria, 30% in Poland and Latvia and more than 75% in the UK and Denmark (Eurostat http://ec.europa.eu/eurostat/statistics-explained/index.php/Main_Page). Despite rapid expansion, there is still room for further development and the e-commerce is rising in impact (Eurostat http://ec.europa.eu/eurostat/statistics-explained/index.php/Main_Page). This impact has already had its effect in the area of postal items distribution. E-commerce, enjoying no limits of time and space, increased the number and volume of international postal items. Postal service providers are required to provide high quality, efficient and reliable services.

Another element, outside the global e-commerce, that affects the postal service market is the emergence of a number of postal and courier service operators and a declining market share of local operators. This process started more than 20 years ago in Western Europe and about 10 years ago in the east European countries.

Rising competition among postal operators has contributed to the development of new types of services, such as parcel lockers, mail tracking systems with real-time tracking information sent to a customer's mobile, etc. Universal service providers have undergone major organizational changes. Currently, RFID tags for item-level tracking are being looked at as an alternative technology to barcodes. The main advantage of this technology is that it is easy to automate data capture process at shipment and sorting hubs.

That said, these changes have failed to solve or even deepened existing problems distributors have to face with respect to the so-called first and last mile (Zellner et al. 2016) in their supply chains. These problems are related to finding the optimal solution in terms of cost and time for both delivery and collection of postal items to/from end customers. And today's definition of a customer may include an individual (home delivery), parcel lockers or a postal outlet.

The main motivation of this research was to build a model for the optimization of postal items distribution (letter and parcel distribution with the use of universal couriers), which would take into account certain currently existing conditions and limitations. The conditions are different from those in the previous models (VRPs, CVRPs, etc.). These new conditions include: a new type of customer with limited capacity (parcel lockers), an opportunity to take alternative delivery points, one point of delivery supported by several couriers, delivery and pick up at the same time and etc. For solving this model, an innovative, original hybrid approach is proposed, which gives a far greater modeling flexibility and efficiency [already tested in modeling and optimization of SCM (Sitek et al. 2014), 2E-CVRP (Sitek 2014) etc.].

The rest of this paper is structured as follows. Section 2 presents the literature review concerning VRPs. Description and formalization of the presented problem is described in Sect. 3. Section 4 explains the proposed implementation approach based on integrated CLP/MP and CLP/heuristic environments. The computational results for numerical experiments are presented in Sect. 5. A summary of conclusions and future works is presented in Sect. 6.

2 Vehicle routing problems: literature review

Various forms of VRPs occur in the field of product distribution/shipments. Solving these problems within acceptable time and optimum costs has become a key issue in modern distribution. In its basic form, a VRP is defined as a problem which answers the question "*What*

is the optimal set of routes for a set of vehicles/fleet/transportation modes to travel in order to deliver to a given set of customers?" This is a combinatorial integer optimization problem (Panos et al. 2013). The VRP objective is generally to minimize route cost. Objective functions could vary with particular variants and applications. The most common objectives are:

- to minimize all transportation cost associated with the used means of transport (e.g. vehicles) as well as the distance travelled;
- to minimize the number of means of transport (e.g. vehicles) needed to cover all delivery points;
- if the estimated travel time and vehicle load have been exceeded, to minimize this overrun.

The VRPs find many applications in real-life but generally concern the services of delivery companies. A detailed description of a VRP is as follows:

- The items are delivered from a depot (one or more).
- A set of vehicles is operated by a set of drivers/couriers.
- Given is the road/train network.
- Given is a set of delivery points where the items are delivered.

The problem is to determine the set of vehicle routes. Each route for each vehicle should start and end at the proper depot. The road network can usually be described using a graph.

Several VRP variants exist (Jairo et al. 2015; Kumar and Panneerselvam 2012; De Jaegere et al. 2014):

- VRPPD (Vehicle Routing Problem with Pickup and Delivery): In this variant, a number of items need to be moved from certain pickup points to other delivery points. The VRPPD objective is to find a set of optimal routes for the means of transport to visit the drop-off and pickup points.
- VRPTW (Vehicle Routing Problem with Time Windows): The delivery points have time windows within which the deliveries must be made.
- CVRP (Capacitated Vehicle Routing Problem): The means of transport have limited carrying capacity. Another CVRP variant, CVRPTW, has time windows.
- 2E-CVRP (Two-Echelon Capacitated Vehicle Routing Problem): Depot-delivery points deliveries pass through intermediate points (called satellites).
- VRPMT (Vehicle Routing Problem with Multiple Trips): The means of transport can do more than one route.
- OVRP (Open Vehicle Routing Problem): The means of transport do not have to return to the depot.
- VRPB (Vehicle Routing Problem with Backhauls):There are two types of delivery points, deliveries (linehauls) and pickups (backhauls). There are additional constraints: (a) each means of transport must perform all the deliveries before making any pickups; (b) routes with only backhauls are disallowed.

There are also many other variants which are derivatives of the above, such as: MT-VRPB (Multiple Trip Vehicle Routing Problem with Backhauls) (Wassan et al. 2016), FSMVRPB (Fleet Size and Mix Vehicle Routing Problem with Backhauls) (Salhi et al. 2013), and VRPBTW (Vehicle Routing Problem with Backhauls and Time Windows) etc.

The most often used methods and approaches for solving VPRs can be divided into exact approaches (mathematical programming, integer programming, etc.), heuristics, constructive methods, two-phase algorithms and metaheuristics (ant algorithms, constraint programming, genetic algorithms, tabu search, simulated annealing etc.) (Yeun et al. 2008; Archetti and Speranza 2014).

The exact methods used for VRPs include mainly the Branch-and-Bound algorithm and its variants e.g., Branch-and-Price, Branch-and-Cost, and Branch-Cut-and-Price. The Branch-Cut-and-Price exact algorithm was used in Ropke et al. (2007) for a CVRP with two-dimensional loading constraints. In Gutiérrez-Jarpa et al. (2010), the exact Branch-and-Price algorithm was proposed for solving a multiple vehicle routing problem with time window—VRPTW. Numerous exact column generation or Dantzig-Wolfe decomposition algorithms were proposed that can accommodate complex constraints and time-dependent costs (Azi et al. 2010; Rousseau et al. 2004). None of those methods has proved capable of dealing with solving VPRs of appreciable size at acceptable time, hence the need to use approximate methods in the form of heuristics and metaheuristics. Among multiple heuristics dedicated to individual VRPs, the most important include the cluster-first, route-second (Fisher and Jaikumar 1981), neighborhood search heuristics to optimize the planned routes of vehicles in the context where new requests, with a pick-up and a delivery location occur in real time (Gandreau et al. 2006), a route construction heuristics with an adaptive parallel scheme (Peng 2011), Granular Tabu Search (GTS) (Toth and Vigo 2014), etc. Nearly all of these methods are heuristic and metaheuristic approaches due to the NP-hardness of the VRP. The Constraint Logic Programming (CLP) (Apt and Wallace 2006; Rossi et al. 2006) approach to many problem types involves building a solution incrementally, backtracking when infeasibility is detected, until a solution is found or the problem is proven to have no solution. This approach can be applied to a variety of VRPs, but it is usually too inefficient to deal with real-world cases. In most real-world cases, the existence of a solution is never in doubt. The quality of the solution is essential and, therefore, CLP is usually used to solve VRP sub-problems or evaluate the feasibility of routes and time slots, available capacity etc. (Rossi et al. 2006).

In some situations, when an exact (optimal) solution is needed, exact methods are used, supported by some heuristics or metaheuristics. Metaheuristics can be applied to finding initial solutions or as presolving methods for narrowing solution spaces or eliminating infeasible solutions. In such cases we can talk about hybrid methods. Motivation behind this work was to construct a reference model for the problem of postal item distribution to final delivery points through universal couriers.

In addition to building the reference model, this contribution includes the implementation and solution of the model with the use of the hybrid approach that applies CLP for initial solving and MP or heuristics for finding the final optimal solution. The universal information structure, based on the sets of facts, was also proposed for the reference model. The structure can be applied to the entire class of VRPs and implemented with ease as databases, XML flat files, etc.

3 Capacitated vehicle routing problem with pick-up and alternative delivery (CVRPPAD): problem description

The presented problem can be described as a variant of VRP with an additional possibility of having means of transport for delivery and pick-up. The capacities of the means of transport and some of the delivery points are taken into account. Delivery points are represented by a postal outlet, an individual customer or a parcel locker with a specified capacity at a given time. The problem assumes alternative delivery points, a characteristic feature of parcel locker locations. Couriers are universal in that they can serve any delivery point type and do pickups and deliveries. The problem can be defined as the Capacitated Vehicle Routing Problem with Pick-up and Alternative Delivery (CVRPPAD). The delivery network for the CVRPPAD is shown in Fig. 1.
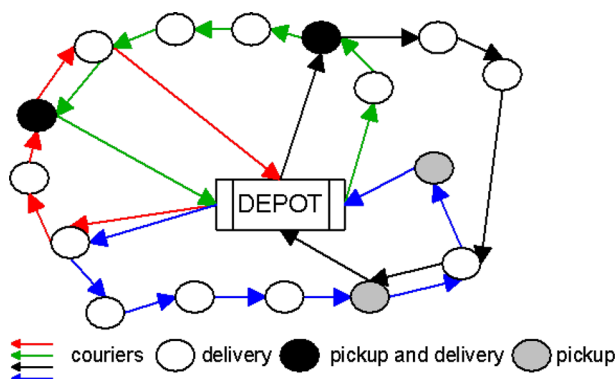
Fig. 1 Schematic diagram of the CVRPPAD

The objective is to minimize the distances for couriers to travel and also the "penalty" for delivering items to alternative points. This is the most obvious objective function chosen for practical reasons, but other options are possible, for example, the number of trips/couriers or the shortest delivery time, etc.

Problem characterization:

- Three categories of delivery recipients/points are considered: i.e., postal outlet (post office, parcel service point, etc.), home delivery (individual customer), parcel locker (PackStation, PACCOMAT, MyQuickBox, etc.)
- The fleet of vehicles is heterogeneous (volume/capacity).
- A vehicle may perform no more than one trip in a planning period.
- Vehicles' capacity constraints are imposed.
- Parcel lockers' capacity constraints are imposed.
- Alternative delivery locations are possible.
- Each route between delivery points has a specified travel time and the overall travel time length cannot be greater than the pre-set travel time length.
- Items cannot reloaded.
- Each courier's route starts from and ends at the depot.
- Note—No constraints are imposed on *route length, number of delivery points on the route, customers' priorities*, however the model is flexible enough to add these constraints if needed.

It has to be emphasized that three of the characteristics above make the formulation of this problem different from the existing VRP variants. The differences are introducing various delivery point types, accounting for the capacity of one delivery point type (parcel lockers) and using alternative delivery points. These three elements are dependent on today's new conditions of delivery services, in particular in urban areas, where delivery points called packer lockers are being introduced (Faugere and Montreuil 2016).

3.1 Problem formulation

Most of VRPs and their variants are modeled as integer linear program (ILP) problems with two-index or rarely with three-index decision variables for the commodity flow (Nagy et al. 2015). According to the assumptions made, the presented problem (Sect. 3) is modeled as a

Table 1 Indices, parameters and decision variables

Symbol	Description
Indices	
P	A set of delivery points /e.g. parcel locker, postal outlet, etc./
I	A set of all items
C	A set of all couriers
p,j	Delivery point index $(p, j \in P)$
i	Item index $(i \in I)$
c	Courier index $(c \in C)$
d	Depot /hub/
Parameters	
vi_i	Item volume (volumetric weight) i $(i \in I)$
vc_c	Courier's vehicle volume c $(c \in C)$
co_p	Number of items that point p can accept $(p \in P)$
de_{ip}	If item i can be delivered to delivery point p, then $de_{ip} = 1$, otherwise $de_{ip} = 0$ $(i \in I, p \in P)$
di_{pj}	Distance between delivery points p, j, $(p, j \in P \cup d)$
pe_{ip}	Penalty for delivery of item i to point p $(i \in I, p \in P)$
ti_{pj}	Transfer time between points p, j, $(p, j \in P \cup d)$
dr_i	If item i is delivered from the depot to the delivery point then $dr_i = 1$ otherwise $dr_i = 0$ $(i \in I)$
T	Non extendible delivery time frames
A	Arbitrarily large constant
Decision variables	
X_{cpj}	If courier c travels from point p to point j, then $X_{cpj} = 1$, otherwise $X_{cpj} = 0$, $(c \in C, p, j \in P \cup d)$
Y_{cpji}	If courier c travels from point p to point j carrying item i, then $Y_{cpji} = 1$, otherwise $Y_{cpji} = 0$, $(c \in C, p, j \in P \cup d)$
E_{ic}	If item i is delivered by courier c, then $E_{ic} = 1$, otherwise $E_{ic} = 0$ $(i \in I, c \in C)$
Z_{ip}	If item i is delivered to the point p, then $Z_{ip} = 1$, otherwise $Z_{ip} = 0$ $(i \in I, p \in P)$

binary linear program (BLP) problem with three-index and four index decision variable formulations. For example, variables X_{cpj} specify whether arc (p, j) is traversed by a particular courier c or not; variables Y_{cpji} specify whether arc (p, j) is traversed by a particular courier c with item i or not. Table 1 shows the indices, parameters and decision variables of the model.

The sum of distances for couriers to travel and the "penalty" for delivering items to alternative delivery points were adopted as the objective function (1). The objective function is minimized.

$$\sum_{c \in C} \sum_{p \in P \cup d} \sum_{j \in P \cup d} di_{pj} \cdot X_{cpj} + \sum_{i \in I} \sum_{p \in P} pe_{ip} Z_{ip} \qquad (1)$$

The model has a set of constraints that define the space of feasible solutions. The proposed set of constraints (2)–(15) satisfies the assumptions made (Sect. 3) and can be regarded as the reference set for a broad class of problems derived from the VRP. Some of these constraints may be removed or adequately parameterized for specific VRP variants. For example, constraint (11) is responsible for the so-called time windows, thereby if T takes a very high value or is removed from the set, the problem will be considered without the time windows, etc. The meanings of the constraints are detailed in Table 2.

Table 2 Meanings of the constraints

Constraints	Description
(2)	Arrival and departure of a courier at/from delivery point (parcel locker, postal outlet, and individual customer)
(3)	If no items are to be carried on the route, a courier does not travel that route
(4)	If a courier does not travel along a route, no items are to be carried on that route
(5)	At no route segment courier carries more items than allowable vehicle volume
(6)	Items are delivered to one of possible delivery points
(7)	One run of a courier
(8)	Item delivered to only one delivery point
(9)	Non extendible delivery point volume/At the point—the number of items matches the number of locations/
(10)	Items picked up/delivered to a delivery point
(11)	Runs executed within the required time
(12)	Each courier picks up/ delivers items from/to a source (depot)
(13)	Assignment of a courier to an item
(14)	A given item can be assigned to one courier
(15)	Binarity

$$\sum_{j\in P\cup d} X_{cpj} = \sum_{j\in P\cup d} X_{cjp} \quad \forall c \in C, \ p \in P \cup d \tag{2}$$

$$X_{cpj} \leq \sum_{i\in I} Y_{cpji} \quad \forall c \in C, \ p \in P, \ j \in P \tag{3}$$

$$Y_{cpji} \leq X_{cpj} \quad \forall c \in C, \ p \in P \cup d, \ j \in P \cup d, \ i \in I \tag{4}$$

$$\sum_{i\in I} vi_i \cdot Y_{cpji} \leq vc_c \quad \forall c \in C, \ p \in P \cup d, \ j \in P \cup d \tag{5}$$

$$\sum_{c\in C} \sum_{p\in P\cup d} Y_{cpji} = Z_{ip} \quad \forall j \in P, \ i \in I{:}de_{ip} = 1, \ dr_i = 1$$

$$\sum_{c\in C} \sum_{p\in P\cup d} Y_{cjpi} = Z_{ip} \quad \forall j \in P, i \in I{:}de_{i,p} = 1, \ dr_i = 0 \tag{6}$$

$$\sum_{j\in P} X_{cdj} \leq 1 \quad \forall c \in C \tag{7}$$

$$\sum_{p\in P} Z_{ip} = 1 \quad \forall i \in I \tag{8}$$

$$\sum_{i\in I} Z_{ip} \leq co_p \quad \forall p \in P \tag{9}$$

$$\sum_{j\in P} \sum_{c\in C} Y_{cjpi} - \sum_{j\in O} \sum_{c\in C} Y_{cpji} = Z_{ip} \quad \forall p \in P, i \in I{:}dr_i = 1$$

$$\sum_{j\in P} \sum_{c\in C} Y_{cjpi} - \sum_{j\in P} \sum_{c\in C} Y_{cpji} = -Z_{ip} \quad \forall p \in P, i \in I{:}dr_i = 0 \tag{10}$$

$$\sum_{p\in P\cup d} \sum_{j\in P\cup d} time_{pj} \cdot X_{cpj} \leq T \quad \forall c \in C \tag{11}$$

$$\sum_{c \in C} \sum_{p \in P} Y_{cdpi} = 1 \quad \forall i \in I : dr_i = 1$$

$$\sum_{c \in C} \sum_{p \in P} Y_{cpdi} = 1 \quad \forall i \in I : dr_i = 0 \tag{12}$$

$$Y_{cpji} = E_{ic} \quad \forall c \in C, p \in P, j \in P, i \in I \tag{13}$$

$$\sum_{c \in C} E_{ic} = 1 \quad \forall i \in I \tag{14}$$

$$Z_{ip} \in \{0, 1\} \quad \forall i \in I, \ p \in P$$

$$Y_{cpji} = \{0, 1\} \quad \forall c \in C, \ p \in P \cup d, \ j \in P \cup d, \ i \in I$$

$$X_{cpj} = \{0, 1\} \quad \forall c \in C, \ p \in P \cup d, \ j \in P \cup d \tag{15}$$

4 Implementation

Due to the character of the proposed model (BLP), mathematical programming (MP) provides a natural environment for its implementation. A number of solvers are available, both commercial (LINGO, CLPEX and free of charge (SCIP, LP_Solve), which use MP methods such as *simplex, branch&bound, branch&price*, etc. (Schrijver 1998). These methods are exact methods, which, taking into account a binary character of decision variables and characteristics of VRPs (NP-hard), renders them inefficient especially in the case of real data instances of larger sizes. Thus based on their previous experience in solving similar problems (Sitek et al. 2014; Sitek 2014; Sitek and Wikarek 2015b), the authors propose a hybrid approach to the implementation and solving (optimization) of the model (1)–(15).

Here, the hybrid approach involves combining two environments, MP and CLP, in the following way. The CLP environment with built-in mechanisms for solving constraints (e.g., constraint propagation, backtracking, etc.) Apt and Wallace (2006) and Rossi et al. (2006) is used in the presolving phase. In this phase, the values of some of the decision variables can be established, the domains of other variables can be narrowed and some constraints can be aggregated and transformed before the actual solving process starts. The presolving phase diagram for the model (1)–(15) is shown in Sect. 4.2 (Fig. 3). The model is solved in the second phase with the use of an MP method. Another element of the implementation is the design of a suitable information structure for the model (Sect. 4.1). The universal information structure allows integrating the model implementation with equivalent commercial environments Enterprise Resource Planning (ERP), Distribution Resources Planning DRP), Warehouse Management System (WMS), Transportation Management System (TMS) and implementing the model with the use of data bases or XML files. The information structure is presented in the form of sets of facts, as in Fig. 2. The hybrid approach with CLP used for presolving is a universal idea and can be combined with other environments outside MP, such as Genetic Algorithm (GA), Ant Colony Optimization ACO, etc.

4.1 Information structure-the sets of facts

The information structure for model (1)–(15) is represented by the sets of facts. A fact is the simplest predicate in the declarative approach (Apt and Wallace 2006). It consists of a single atomic expression and all of its arguments are constants. It can be interpreted in the way a relation is interpreted in a relational approach. The fact can be represented by, for example: *Person(ID,forename,surname,age), Vehicle(Registration_numbe, Capacity,*
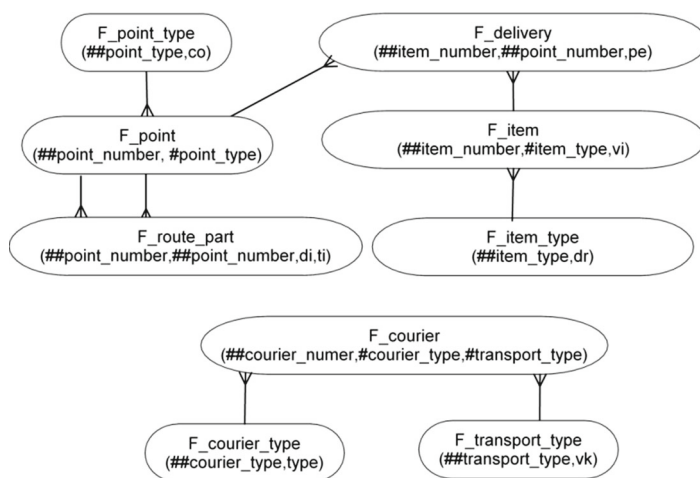
Fig. 2 The structure and relationships between set of facts for reference model

Production_date), etc. It has key and descriptive attributes. The key attributes are used to determine the instance of a given fact. It is relatively easy to transform a set of facts into a set of tables in the relational database or a set of XML files. Fig. 2 shows a diagram of the structure of facts and their relationships.

4.2 Presolving

Presolving is used to reduce the dimensions of the model and, consequently, provides a possibility of solving other large-size models within acceptable time. Figure 3 shows the diagram outlining the presolving stages for the (1)–(15) model. In the first stage, the facts *F_item, F_delivery* are transformed. All postal items can be grouped according to a category, e.g., a delivery point and an item type, which reduces the number of postal items to handle (groups instead of single postal items are considered). In the next stage, parameter (de_{ip}) is used to establish which items (groups) have a delivery point alternative. After the analysis of this variable, some of decision variables Z_{ip} take the value of 0 or 1, whereas the remaining variables that can take the value 1 and 0 are not established at this stage (their values are established during the optimization stage). On the basis of constraints (6) and (12), the values of those decision variables Y_{cpji} for which Z_{ip} have been established in the earlier stage are determined. The final presolving stage involves determining the values of X_{cpj} on the basis of constraints (3) and (4) and decision variables Y_{cpji} of pre-established values.

Obviously, the presolving technique allowed determining, before the actual solving process (optimization), the values of some decision variables on the basis of facts and some constraints, thereby reducing the number of variables for the ultimate solving process (Table 4a, b, Sect. 5). In practice, particular presolving stages were executed with the use of adequate CLP predicates. This presolving method implemented using the CLP can be hybridized with other methods, both exact (Yeun et al. 2008) and metaheuristics (Archetti and Speranza 2014).
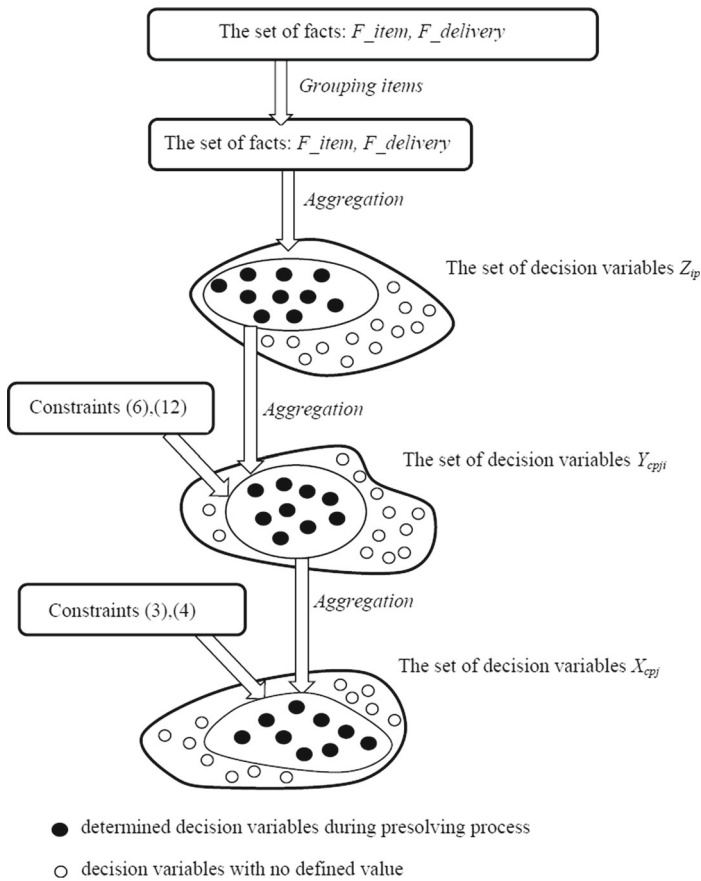
Fig. 3 The schema of presolving phase for proposed model

4.3 Heuristic for solving CVRPPAD

For the problems of larger sizes, considering their combinatorial character, the proposed hybrid approach which integrates MP and CLP is insufficiently effective. Therefore, a heuristic is proposed (Fig. 4), which can replace MP in the hybrid approach. The heuristic is dedicated to the structure and properties of problems such as CVRPPAD. It does not guarantee finding an optimal solution, but in combination with CLP, it provides an effective and efficient tool. The heuristic is characterized by three selection criteria/priorities for *R1, R2, R3* parameters (Table 3). Being very flexible, this approach can be used according to the character of data. The travelling salesman problem was solved with the known from the literature variants of NN (Nearest Neighborhood) methods (Gandreau et al. 2006; Peng 2011; Regoa et al. 2011).

In short, the main idea of the heuristic algorithm (Fig. 4) is that in the first step, a courier is chosen based on the R1 criterion and the deliveries and their recipients are determined based on criteria R2 and R3. In the next step, the route for the courier is set (in accordance with the NN method).

The heuristic can be used for solving CVRPPAD in two ways, either directly or as an element of the hybrid approach, in which it will replace the MP solver. Its unquestionable
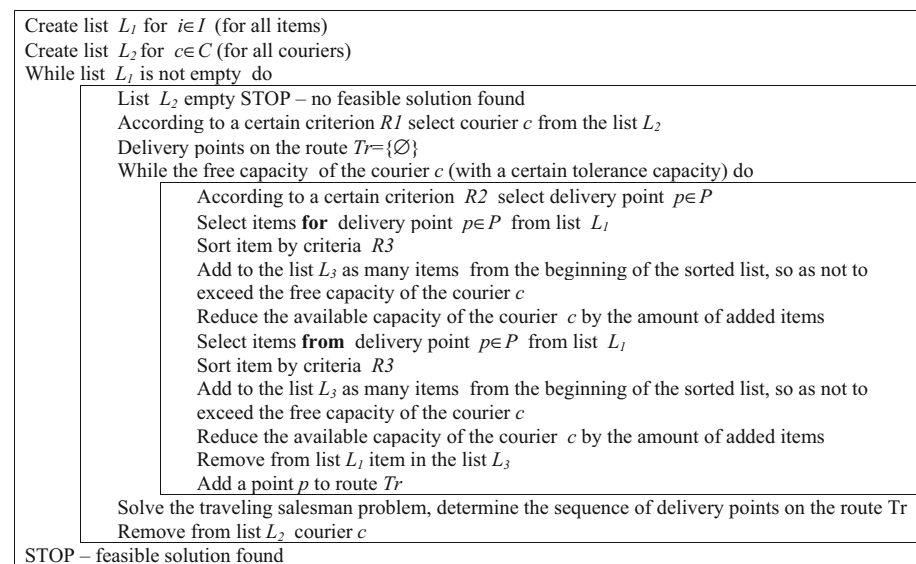
Create list L_1 for $i \in I$ (for all items)
Create list L_2 for $c \in C$ (for all couriers)
While list L_1 is not empty do
        List L_2 empty STOP – no feasible solution found
        According to a certain criterion $R1$ select courier c from the list L_2
        Delivery points on the route $Tr = \{\varnothing\}$
        While the free capacity of the courier c (with a certain tolerance capacity) do
                According to a certain criterion $R2$ select delivery point $p \in P$
                Select items **for** delivery point $p \in P$ from list L_1
                Sort item by criteria $R3$
                Add to the list L_3 as many items from the beginning of the sorted list, so as not to exceed the free capacity of the courier c
                Reduce the available capacity of the courier c by the amount of added items
                Select items **from** delivery point $p \in P$ from list L_1
                Sort item by criteria $R3$
                Add to the list L_3 as many items from the beginning of the sorted list, so as not to exceed the free capacity of the courier c
                Reduce the available capacity of the courier c by the amount of added items
                Remove from list L_1 item in the list L_3
                Add a point p to route Tr
        Solve the traveling salesman problem, determine the sequence of delivery points on the route Tr
        Remove from list L_2 courier c
STOP – feasible solution found

Fig. 4 The algorithm for heuristic dedicated to the problem CVRPPAD

Table 3 Possible values of criterion

Criterion	Possible values
R1	Courier with the highest capacity
	Courier the smallest capacity
R2	Delivery point which has the largest capacity
	Delivery point which has the smallest capacity
	Delivery point with the largest number of items
	Delivery point with the smallest number of items
	Delivery point with the largest capacity of items
	Delivery point with the smallest capacity of items
R3	First, the items that do not have an alternative delivery points
	First, the items with bigger capacity from the delivery point

The criteria underlines were chosen for computational experiments (Sect. 5)

advantage consists of the reduced computation time and the possibility of solving the models of larger sizes within acceptable time.

The heuristic used for solving CVRPPAD yields optimal or approximate (from 0 to 150% worse than optimal) solutions. When used in the hybrid approach, it reduces the computation time considerably and improves the quality of approximate solutions (0–10% worse than optimal) (Sect. 5 Table 4b).

5 Numerical experiments

In order to verify the proposed model as well as evaluate the hybrid approach, we made a number of computational experiments (P1…P12) with instances of the facts contained in

Table 4 Results for numerical experiments P1…P12

P	Lpr	Lpu	Lku	MP-based approach				Hybrid1 approach (CLP and MP)			
				V	C	T	F_C	V	C	T	F_C
(a)											
MPsolver-LINGO											
P1	20	5	2	1672	404	234	128	580	294	4	128
P2	20	5	2	1672	404	454	39	580	294	7	39
P3	20	5	2	1672	404	346	139	580	294	10	139
P4	20	8	4	5243	872	87	230	2946	2124	45	230
P5	40	8	4	10443	1283	345	258	2946	2124	67	258
P6	60	8	4	15543	1523	600**	258*	2946	2124	36	258
P7	80	8	4	20642	1813	600**	NFSF	2946	2124	39	262
P8	100	8	4	25743	2093	600**	NFSF	2946	2124	42	262
P9	1000	100	20	35674567	20056	2400**	NFSF	267804	456747	987	56455
P10	1000	200	20	56675643	45567	2400**	NFSF	456456	774564	1545	97865
P11	2000	100	20	75434534	56576	2400**	NFSF	267804	456747	878	56455
P12	2000	200	20	143456467	87565	2400**	NFSF	456456	774564	2134	104343
MPsolverSCIP											
P1	20	5	2	1672	404	185	128	580	294	6	128
P2	20	5	2	1672	404	302	39	580	294	8	39
P3	20	5	2	1672	404	245	139	580	294	9	139
P4	20	8	4	5243	872	81	230	2946	2124	41	230
P5	40	8	4	10443	1283	301	258	2946	2124	54	258
P6	60	8	4	15543	1523	574	258	2946	2124	31	258
P7	80	8	4	20642	1813	600**	456*	2946	2124	39	262
P8	100	8	4	25743	2093	600**	NFSF	2946	2124	41	262
P9	1000	100	20	35674567	20056	2400**	NFSF	267804	456747	876	56455
P10	1000	200	20	56675643	45567	2400**	NFSF	456456	774564	1321	97865
P11	2000	100	20	75434534	56576	2400**	NFSF	267804	456747	794	56455
P12	2000	200	20	143456467	87565	2400**	NFSF	456456	774564	1934	104343

P	Lpr	Lpu	Lku	Heuristics		Hybrid2 approach (CLP and Heuristics)	
				T	F_C	T	F_C
(b)							
P1	20	5	2	23	128	12	128
P2	20	5	2	23	42*	12	39
P3	20	5	2	23	154*	12	139
P4	20	8	4	23	260*	12	230
P5	40	8	4	26	324*	13	260*
P6	60	8	4	28	258	13	258
P7	80	8	4	28	294*	15	274*
P8	100	8	4	29	276*	14	286*
P9	1000	100	20	47	67434*	22	56455

Table 4 continued

P	Lpr	Lpu	Lku	Heuristics		Hybrid2 approach (CLP and Heuristics)	
				T	F_C	T	F_C
P10	1000	200	20	49	124332*	22	97967*
P11	2000	100	20	53	67531*	22	56674*
P12	2000	200	20	52	264563*	24	104343

Pi   Example Id
C   The number of constraints
F_c The optimal value of objective function
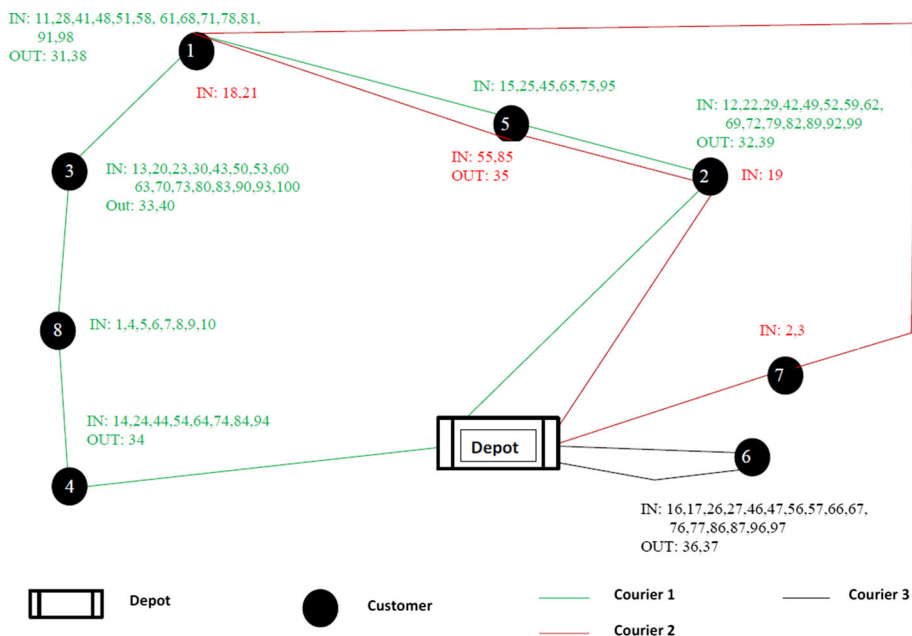Lpr   The number of items
V   The number of decision variables
T   Computational time in seconds
**   Calculations interrupted after time 600s/2400s
Lpu The number of points
*   The feasible value of objective function
Lku   The number of couriers

Fig. 5 The optimal solution for P8 (no time windows, the longest route 28) Fc = 262 (two items in alternative places). IN-delivery, OUT-pick up

Appendix A (due to the limited volume of the paper, complete data instances were included only for P1…P8). The experiments varied in the number of couriers (up to 20 couriers), delivery points (5–200), items (up to 2000), types of transport (up to 3 modes of transport), and in the type of delivery points (2).

To implement the models and carry out computational experiments, Eclipse (www.eclipse.org) was used as part of the CLP, while Lingo (www.lindo.com) and (http://scip.zib.de/) were used as MP solvers.
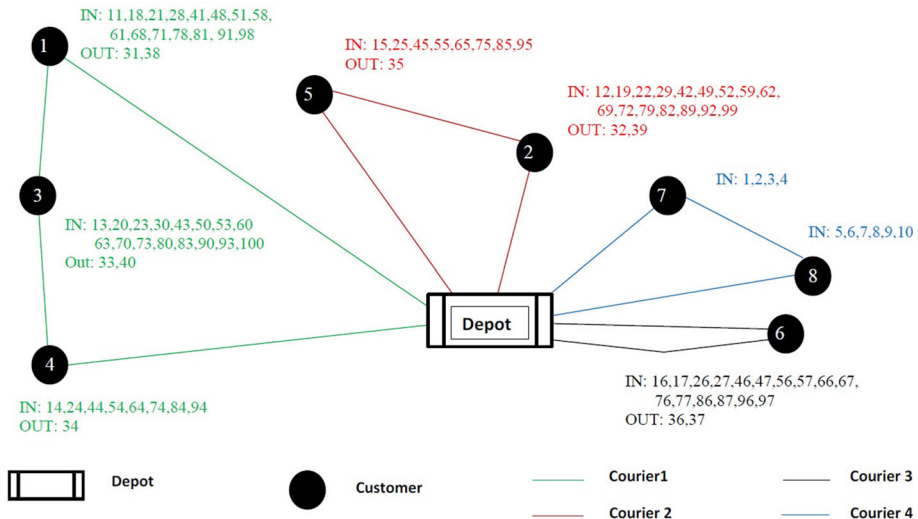
Fig. 6 The feasible solution for P8 (no time windows, the longest route 28) Fc = 470 (four items in alternative places). IN-delivery, OUT-pick up

In the first stage of the experiment, examples P1…P12 were implemented and solved in four approaches, MP, hybrid1 (CLP/MP with presolving) heuristics and hybrid2 (CLP/Heuristics with presolving), with no time windows included. The results are shown in Table 4a, b and the detailed results for P1, P2, P3 and P8 in Table 5 (Appendix B). The optimal solution (hybrid-based approach) for P8 is shown in Fig. 5 and the feasible solution (MP-based approach) in Fig. 6 and in Table 6 (Appendix B). From these data, it follows that the use of the hybrid1/hybrid2 approach and heuristics made it possible to find a solution in a much shorter time.

Analysis of the results clearly shows the superiority of hybrid approach. The hybrid approach (hybrid1) as an exact method reduces calculation time by the order of 2 to 65 times compared to the MP method. However, using different solvers, MP is not so significant. SCIP is approximately 20% more effective than LINGO.

The hybrid approach (hybrid2) provides double reduction in the computation time and improves the quality of approximate solutions (0–10% worse than optimal) in relation to heuristics (the quality of approximate solutions are 0–150% worse than optimal).

In the second stage of the experiment, time windows were implemented for example P8. The optimal solution is shown in Fig. 7 and in Table 7 (Appendix B). The proposed model provides daily decision support in planning the routes of couriers with minimal costs, optimal allocating items to the couriers, determining daily requirement for couriers, etc.

6 Conclusions

This paper introduced a new VRP variant called CVRPPAD (Capacitated Vehicle Routing Problem with Pick-up and Alternative Delivery), a crucial tool for postal items distribution, city logistic, etc. The CVRPPAD has unique features that distinguish it from other variants of the VRPs (Sect. 3). The most important characteristics include taking into account the capacity of delivery points and alternative locations. These changes are due to the inclusion
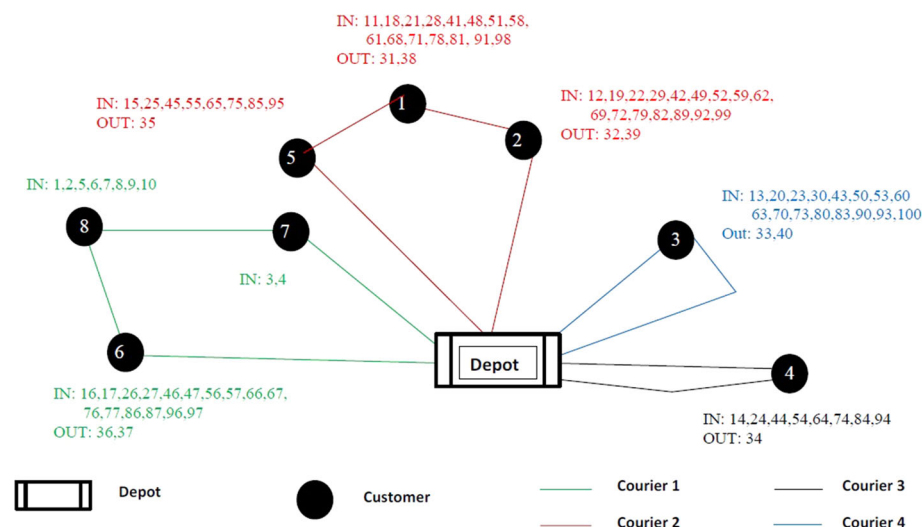
Fig. 7 The optimal solution for P8 with time window (the longest route ≤ 20) Fc = 470 (two items in alternative places). IN-delivery, OUT-pick up

of a new type of delivery point—a parcel locker. The proposed model of the CVRPPAD combines the features of many different variants of VRP such as delivery and pick-ups, vehicle capacity, different type of vehicles and so on.

The model was tested with the classical MP environment, "LINGO" and "SCIP", with the proposed hybrid approach combining two environments, MP and CLP (Bockmayr and Kasper 2004; Milano and Wallace 2010; Sitek and Wikarek 2016; Hooker 2002; Sitek and Wikarek 2015a; Silva 2001), and with hybrid approach combining CLP and proposed heuristics (Sect. 4.3). Declarative CLP was used to presolve the model. MP and heuristics were used for the final model solving (optimization). The use of this integrated approach has reduced the size of the model by the factor of almost 10, thus reducing significantly the optimization time and, for the larger examples, finding the solution within acceptable time span, as opposed to the MP implementations. The proposed hybrid approach, the presolving method in particular, can be used for the integration with heuristics and metaheuristics necessary to solve the CVRPPAD model of industrial size.

The proposed model and method of its solution can be widely implemented in information management systems of postal items distribution. For this purpose, a universal information structure is proposed in the form of sets of facts that are easily integrated with the databases. Further research will follow two directions.

The model will be expanded to cover a route length, number of points on the route, items prioritization, lead times etc. (Nielsen et al. 2014b; Nielsen et al. 2014a) and implementation in the cloud (Deniziak et al. 2013). The model is planned to be enriched with fuzzy logic (Bocewicz et al. 2016).

The authors declare that there is no conflict of interests regarding the publication of this paper.

Appendix A: Data instances in the form of sets of facts -for numerical experiments

```
F_point_type(##point_type, the maximum number of items).
 Example (P1,P2,P4,P5,P6,P7,P8): F_point_type(1,1000). F_point_type(2,4).
 Example (P3): F_point_type(1,1000). F_point_type(2,5).
F_point(##point_number, #point_type)
 Example (P1,P2,P3,P4,P5,P6,P7,P8): F_point(1,1). F_point(2,1). F_point(3,1).
 Example (P1,P2,P3): F_point(4,2). F_point(5,2).
 Example (P4,P5,P6,P7,P8): F_point(4,1). F_point(5,1). F_point(6,1).
  F_point(7,2). F_point(8,2).
F_item_type(##item_type, delivered/picked up, the maximum volume/ weight)
 Example (P1,P2,P3,P4,P5,P6,P7,P8): F_item_type(1,"delivered",10).
  F_item_type(2," picked up",10)
F_item(##item_number, #item_type, the volume/weight)
 Example (P1,P2,P3,P4,P5,P6,P7,P8): F_item(1,1,2). F_item(2,1,2). F_item(3,1,2).
  F_item(4,1,2). F_item(5,1,2). F_item(6,1,2). F_item(7,1,2).  F_item(8,1,2).
  F_item(9,1,2). F_item(10,1,2). F_item(11,2,1). F_item(12,2,1).
 Example (P1, P2, P3): F_item(13,2,1). F_item(14,2,1). F_item(15,2,1).
  F_item(16,2,1). F_item(17,2,1). F_item(18,2,1). F_item(19,2,2).
  F_item(20,2,2).
 Example (P4, P5, P6, P7): F_item(13,1,1). F_item(14,1,1). F_F_item(15,1,1).
  F_item(16,1,1). F_item(17,1,1). F_item(18,1,1). F_item(19,1,1).
  F_item(20,1,1). F_item(21,1,1). F_item(22,1,1). F_item(23,1,1).
  F_item(24,1,1). F_item(25,1,1). F_item(26,1,1). F_item(27,1,1).
  F_item(28,1,1). F_item(29,1,1). F_item(30,1,1). F_item(31,2,1).
  F_item(32,2,1). F_item(33,2,1). F_item(34,2,1). F_item(35,2,1).
  F_item(36,2,1). F_item(37,2,1). F_item(38,2,1). F_item(39,2,1).
  F_item(40,2,1). F_item(41,1,1). F_item(42,1,1). F_item(43,1,1).
  F_item(44,1,1). F_item(45,1,1). F_item(46,1,1). F_item(47,1,1).
  F_item(48,1,1). F_item(49,1,1). F_item(50,1,1). F_item(51,1,1).
  F_item(52,1,1). F_item(53,1,1). F_item(54,1,1). F_item(55,1,1).
  F_item(56,1,1). F_item(57,1,1). F_item(58,1,1). F_item(59,1,1).
  F_item(60,1,1). F_item(61,1,1). F_item(62,1,1). F_item(63,1,1).
  F_item(64,1,1). F_item(65,1,1). F_item(66,1,1). F_item(67,1,1).
  F_item(68,1,1). F_item(69,1,1). F_item(00,1,1). F_item(71,1,1).
  F_item(72,1,1). F_item(73,1,1). F_item(74,1,1). F_item(75,1,1).
  F_item(76,1,1). F_item(77,1,1). F_item(78,1,1). F_item(79,1,1).
  F_item(80,1,1). F_item(81,1,1). F_item(82,1,1). F_item(83,1,1).
  F_item(84,1,1). F_item(85,1,1). F_item(86,1,1). F_item(87,1,1).
  F_item(88,1,1). F_item(89,1,1). F_item(90,1,1). F_item(91,1,1).
  F_item(92,1,1). F_item(93,1,1). F_item(94,1,1). F_item(95,1,1).
  F_item(96,1,1). F_item(97,1,1). F_item(98,1,1). F_item(99,1,1).
  F_item(100,1,1).
F_delivery(##item_number, ##point_number, penalty for delivery).
 Example (P1, P2, P3):F_delivery(1,4,0).  F_delivery(2,4,0). F_delivery(3,4,0).
  F_delivery(4,4,0).  F_delivery(5,4,0).  F_delivery(6,1,0). F_delivery(7,2,0).
  F_delivery(8,3,0). F_delivery(9,1,0).  F_delivery(10,2,0).F_delivery(11,3,0).
  F_delivery(12,1,0). F_delivery(13,2,0). F_delivery(14,3,0).F_delivery(15,1,0).
  F_delivery(16,2,0). F_delivery(17,3,0). F_delivery(18,1,0).F_delivery(19,2,0).
  F_delivery(20,3,0). F_delivery(1,5,100).F_delivery(2,5,100).
  F_delivery(3,5,100).F_delivery(4,5,100).F_delivery(5,5,100).
 Example (P4, P5, P6, P7): F_delivery(1,8,0). F_delivery(2,8,0).
  F_delivery(3,8,0). F_delivery(4,8,0). F_delivery(5,8,0). F_delivery(6,8,0).
  F_delivery(7,8,0). F_delivery(8,8,0). F_delivery(9,8,0). F_delivery(10,8,0).
  F_delivery(11,1,0).F_delivery(12,2,0).F_delivery(13,3,0).F_delivery(14,4,0).
  F_delivery(15,5,0).F_delivery(16,6,0).F_delivery(17,6,0).F_delivery(18,1,0).
  F_delivery(19,2,0).F_delivery(20,3,0).F_delivery(21,1,0).F_delivery(22,2,0).
  F_delivery(23,3,0).F_delivery(24,4,0).F_delivery(25,5,0).F_delivery(26,6,0).
  F_delivery(27,6,0).F_delivery(28,1,0).F_delivery(29,2,0).F_delivery(30,3,0).
  F_delivery(31,1,0).F_delivery(32,2,0).F_delivery(33,3,0).F_delivery(34,4,0).
  F_delivery(35,5,0).F_delivery(36,6,0).F_delivery(37,6,0).F_delivery(38,1,0).
  F_delivery(39,2,0).F_delivery(40,3,0).F_delivery(41,1,0).F_delivery(42,2,0).
```

```
    F_delivery(43,3,0).F_delivery(44,4,0).F_delivery(45,5,0).F_delivery(46,6,0).
    F_delivery(47,6,0).F_delivery(48,1,0).F_delivery(49,2,0).F_delivery(50,3,0).
    F_delivery(51,1,0).F_delivery(52,2,0).F_delivery(53,3,0).F_delivery(54,4,0).
    F_delivery(55,5,0).F_delivery(56,6,0).F_delivery(57,6,0).F_delivery(58,1,0).
    F_delivery(59,2,0).F_delivery(60,3,0).F_delivery(61,1,0).F_delivery(62,2,0).
    F_delivery(63,3,0).F_delivery(64,4,0).F_delivery(65,5,0).F_delivery(66,6,0).
    F_delivery(67,6,0).F_delivery(68,1,0).F_delivery(69,2,0).F_delivery(70,3,0).
    F_delivery(71,1,0).F_delivery(72,2,0).F_delivery(73,3,0).F_delivery(74,4,0).
    F_delivery(75,5,0).F_delivery(76,6,0).F_delivery(77,6,0).F_delivery(78,1,0).
    F_delivery(79,2,0).F_delivery(80,3,0).F_delivery(81,1,0).F_delivery(82,2,0).
    F_delivery(83,3,0).F_delivery(84,4,0).F_delivery(85,5,0).F_delivery(86,6,0).
    F_delivery(87,6,0).F_delivery(88,1,0).F_delivery(89,2,0).F_delivery(90,3,0).
    F_delivery(91,1,0).F_delivery(92,2,0).F_delivery(93,3,0).F_delivery(94,4,0).
    F_delivery(95,5,0).F_delivery(96,6,0).F_delivery(97,6,0).F_delivery(98,1,0).
    F_delivery(99,2,0).F_delivery(100,3,0).F_delivery(1,7,100).
    F_delivery(2,7,100).F_delivery(3,7,100).F_delivery(4,7,100).
    F_delivery(5,7,100).F_delivery(6,7,100).F_delivery(7,7,100).
    F_delivery(8,7,100).F_delivery(9,7,100).F_delivery(10,7,100).
F_courier_type(##courier_type, type)
    Example (P1,P2,P3,P4,P5,P6,P7,P8): F_courier_type(1,"Universal").
F_transport_type(##transport_type, transport capacity/volume/weight)
    Example (P1) :F_transport_type(1,50). F_transport_type(2,40).
    Example (P2,P3) : F_transport_type(1,15). F_transport_type(2,18).

    Example (P4, P5, P6, P7, P8) : F_transport_type(1,80). F_transport_type(2,70).
     F_transport_type(3,50).
F_courier(##courier_numer, #courier_type, #transport_type)
    Example (P1, P2, P3): F_courier(1,1,1). F_courier(2,1,2).
    Example (P4, P5, P6, P7,P8): F_courier(1,1,1). F_courier(2,1,2).
     F_courier(3,1,3). F_courier(4,1,3)
F_route_part(##point_number, ##point_number, distance, time)
    Example (P1,P2,P3,P4,P5,P6,P7,P8): F_route_part(d,d,0,0).
     F_route_part(d,1,8,8). F_route_part(d,2,6,6). F_route_part(d,3,10,10).
     F_route_part(d,4,9,9). F_route_part(d,5,7,7). F_route_part(1,d,8,8).
     F_route_part(1,1,0,0). F_route_part(1,2,2,2). F_route_part(1,3,3,3).
     F_route_part(1,4,9,9). F_route_part(1,5,1,1). F_route_part(2,d,6,6).
     F_route_part(2,1,2,2). F_route_part(2,2,0,0). F_route_part(2,3,4,4).
     F_route_part(2,4,9,9). F_route_part(2,5,1,1). F_route_part(3,d,10,10).
     F_route_part(3,1,3,3). F_route_part(3,2,4,4). F_route_part(3,3,0,0).
     F_route_part(3,4,8,8). F_route_part(3,5,3,3). F_route_part(4,d,9,9).
     F_route_part(4,1,9,9). F_route_part(4,2,9,9). F_route_part(4,3,8,8).
     F_route_part(4,4,0,0). F_route_part(4,5,8,8). F_route_part(5,d,7,7).
     F_route_part(5,1,1,1). F_route_part(5,2,1,1). F_route_part(5,3,3,3).
     F_route_part(5,4,8,8). F_route_part(5,5,0,0).
    Example (P4, P5, P6, P7, P8): F_route_part(d,6,4,4). F_route_part(d,7,8,8).
     F_route_part(d,8,10,10). F_route_part(1,6,5,5).    F_route_part(1,7,4,4).
     F_route_part(1,8,6,6).    F_route_part(2,6,3,3).    F_route_part(2,7,4,4).
     F_route_part(2,8,6,6).    F_route_part(3,6,6,6).    F_route_part(3,7,3,3).
     F_route_part(3,8,4,4).    F_route_part(4,6,7,7).    F_route_part(4,7,5,5).
     F_route_part(4,8,4,4).    F_route_part(5,6,4,4).    F_route_part(5,7,3,3).
     F_route_part(5,8,5,5).    F_route_part(6,d,4,4).    F_route_part(6,1,5,5).
     F_route_part(6,2,3,3).    F_route_part(6,3,6,6).    F_route_part(6,4,7,7).
     F_route_part(6,5,4,4).    F_route_part(6,6,0,0).    F_route_part(6,7,4,4).
     F_route_part(6,8,6,6).    F_route_part(7,d,8,8).    F_route_part(7,1,4,4).
     F_route_part(7,2,4,4).    F_route_part(7,3,3,3).    F_route_part(7,4,5,5).
     F_route_part(7,5,3,3).    F_route_part(7,6,4,4).    F_route_part(7,7,0,0).
     F_route_part(7,8,2,2).    F_route_part(8,d,10,10). F_route_part(8,1,6,6).
     F_route_part(8,2,6,6).    F_route_part(8,3,4,4).    F_route_part(8,4,4,4).
     F_route_part(8,5,5,5).    F_route_part(8,6,6,6).    F_route_part(8,7,2,2).
     F_route_part(8,8,0,0).
```

Appendix B: The detailed results for numerical experiments

Table 5 The detailed results for numerical experiments P1, P2, P3, P8

			Example P1	
c	p	j	c	i
1	d	2	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20
1	2	5		
1	5	1	p	i
1	1	3	4	1,2,3,4,5
1	3	4	The rest of the items had no alternative points of delivery	
1	4	d		
			Example P2	
c	p	j	c	i
1	d	2	1	1, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20
1	2	5	2	2, 3, 4, 5
1	5	3		
1	3	1	p	i
1	1	d	4	2,3,4,5
2	d	4	5	1
2	4	d	The rest of the items had no alternative points of delivery	
			Example P3	
c	p	j	c	i
1	d	4	1	1, 2, 3, 4, 5
1	4	d	2	6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20
2	d	2		
2	2	3	p	i
2	3	1	4	1,2,3,4,5
2	1	d	The rest of the items had no alternative points of delivery	
			Example P8	
c	p	j	c	i
1	d	4	1	1,4,5,6,7,8,9,10,11,12,13,14,15,20,22,23,25,28,29,30,31,32,33,34,38,
1	4	8		39,40,41,42,43,44,45,48,49,50,51,52,53,54,58,59,60,62,63,64,65,68
1	8	3		69,70,71,72,73,74,75,78,79,80,81,82,83,84,88,89,90,91,92,93,94,95
1	3	1		98,99,100
1	1	5	2	2,3,18,19,21,24,35,55,61,85
1	5	2	3	16,17,26,27,36,37,46,47,56,57,66,67,76,77,86,87,96,97
1	2	d		
2	d	2	p	i
2	2	5	7	2,3
2	5	1	8	1,4,5,6,7,8,9,10
2	1	7	The rest of the items had no alternative points of delivery	
2	7	4		
2	4	d		
3	d	6		
3	6	d		

Table 6 Example P8: feasible solution

c	p	j		c	i
1	d	4		1	11,13,14,18,20,21,23,24,28,30,31,33,34,38,40,41,43,44,48,
1	4	3			50,51,53,54,58,60,61,63,64,68,70,71,74,81, 73,80,83,84,88,
1	3	1			90,91,93,94,98,100
1	1	d		2	12,15,19,22,25,29,32,35,3942,45,49,52,55,59,62,65,69,72,75,
2	d	5			79,82,85,89,92,95,99
2	5	2		3	16,17,26,27,36,37,46,47,56,57,66,67,76,77,86,87,96,97
2	2	d		4	1,2,3,4,5,6,7,8,9,10
3	d	6			
3	6	d		p	i
4	d	7		7	1,2,3,4
4	7	8		8	5,6,7,8,9,10
4	8	d		The rest of the items had no alternative points of delivery	

Table 7 Example P8: optimal solution with time window $T \leq 20$

c	p	j		c	i
1	d	6		1	1,2,3,4,5,6,7,8,9,10,
1	6	8			16,17,26,27,46,47,56,57,66,67,76,77,86,87,96,97
1	8	7		2	11,12,15,18,19, 21,22,25,28,29,31,32,35,38,39,41,
2	d	5			42,45,48,49,51,52,55,58,59,61,62,65,68,
2	5	1			69,71,72,75,78,79,81,82,85,89,91,92,95,98,99
2	1	2		3	13,20,23,30,33,40,43,50,53,60,63,70,73,80,83,90,93,100
2	2	d		4	14,24,34,44,54,64,74,84,94
3	d	3			
3	3	d		p	i
4	d	4		7	3,4
4	4	d		8	1,2,5,6,7,8,9,10
				The rest of the items had no alternative points of delivery	

References

Apt, K., & Wallace, M. (2006). *Constraint logic programming using eclipse*. Cambridge: Cambridge University Press.

Archetti, C., & Speranza, M. G. (2014). A survey on matheuristics for routing problems. *EURO Journal on Computational Optimization*,. https://doi.org/10.1007/s13675-014-0030-7.

Azi, N., Gendreau, M., & Potvin, J. Y. (2010). An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles. *European Journal of Operational Research*, *202*(3), 756–763. https://doi.org/10.1016/j.ejor.2009.06.034.

Bocewicz, G., Nielsen, I., & Banaszak, Z. (2016). Production flows scheduling subject to fuzzy processing time constraints. *International Journal Of Computer Integrated Manufacturing*, *29*(10), 1105–1127. https://doi.org/10.1080/0951192X.2016.1145739.

Bockmayr, A., & Kasper, T. (2004). Branch-and-infer, a framework for combining CP and IP. *Constraint and Integer Programming Operations Research/Computer Science Interfaces Series*, *27*, 59–87.

De Jaegere, N., Defraeye, M., & Van Nieuwenhuyse, I. (2014). The vehicle routing problem: State of the art classification and review. In: FEB research report KBI_1415, FEB research report KBI_1415.

de Silva, A. (2001). Combining constraint programming and linear programming on an example of bus driver scheduling. *Annals of Operations Research*, *108*, 277. https://doi.org/10.1023/A:1016075431943.

Deniziak, R., Bąk, S., & Czarnecki, R. (2013). Synthesis of real-time cloud applications for internet of things. *Turkish Journal of Electrical Engineering Computer Sciences*, *7719*, 35–49.

Eclipse—the eclipse foundation open source community website, www.eclipse.org. Accessed 20 Jan 2016.

Eurostat - Statistics Explained - Europa.eu. http://ec.europa.eu/eurostat/statistics-explained/index.php/Main_Page. Accessed 23 June (2016).

Faugere, L., & Montreuil, B. (2016). Hyperconnected City logistics: Smart Lockers Terminals and last mile delivery networks. In: *3rd international physical internet conference*, Atlanta, GA USA.

Fisher, M., & Jaikumar, R. (1981). A generalized assignment heuristic for vehicle routing. *Networks*, *11*(2), 109–124. https://doi.org/10.1002/net.3230110205.

Gandreau, M., Guertin, F., Potvin, J. Y., & Seguin, R. (2006). Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries. *Transportation Research Part E: Logistics and Transportatio*, *14*(3), 157–174.

Gutiérrez-Jarpa, G., Desaulniers, G., Laporte, G., & Marianov, V. (2010). A branch-and-price algorithm for vehicle routing problem with deliveries, selective pickups and time windows. *European Journal of Operational Re- search*, *206*(12), 341–349. https://doi.org/10.1016/j.ejor.2010.02.037.

Hooker, J. N. (2002). Logic optimization, and constraint programming. *Journal on Computing*, *14*(4), 295–321.

Jairo, R., Montoya, T., Francob, J. L., Isazac, S. N., Jiménezd, H. F., & Herazo-Padillae, N. (2015). A literature review on the vehicle routing problem with multiple depots. *Computers and Industrial Engineering*, *79*, 115–129.

Kumar, S. N., & Panneerselvam, R. (2012). A survey on the vehicle routing problem and its variants. *Intelligent Information Management*, *2012*(4), 66–74.

Lindo Systems INC, LINDO^{TM}, www.lindo.com. Accessed 20 Jan 2016.

Milano, M., & Wallace, M. (2010). Integrating operations research in constraint programming. *Annals of Operations Research*, *175*, 37–76. https://doi.org/10.1007/s10479-009-0654-9.

Nagy, G., Wassan, N. A., Speranza, M. G., & Archetti, C. (2015). The vehicle routing problem with divisible deliveries and pickups. *Transportation Science*, *49*(2), 271–294. https://doi.org/10.1287/trsc.2013.0501.

Nielsen, P., Jiang, L., Rytter, N. G. M., & Chen, G. (2014a). An investigation of forecast horizon and observation fit's influence on an econometric rate forecast model in the liner shipping industry. *Maritime Policy and Management*, *41*(7), 667–682. https://doi.org/10.1080/03088839.2014.960499.

Nielsen, P., Michna, Z., & Do, N. A. D. (2014b). An empirical investigation of lead time distributions. *IFIP Advances in Information and Communication Technology*, *438*(1), 435–442. https://doi.org/10.1007/978-3-662-44739-0_53.

Panos, M., Pardalos, D., & Graham, R. L. (2013). *Handbook of combinatorial optimization*. Berlin: Springer. ISBN: 978-1-4419-7996-4.

Peng, K. W. (2011). An adaptive parallel route construction heuristic for the vehicle routing problem with time windows constraints. *Expert Systems with applications*, *38*(9), 11939–11946.

Regoa, C., Gamboab, D., Gloverc, F., & Ostermana, C. (2011). Traveling salesman problem heuristics: Leading methods, implementations and latest advances. *European Journal of Operational Research*, *211*(3), 427–441. https://doi.org/10.1016/j.ejor.2010.09.010.

Ropke, S., Cordeau, J. F., Vigo, D., & Iori, M. (2007). Branch-and-cut-and-price for the capacitated vehicle routing problem with two-dimensional constraints. In: *Proceedings of ROUTE*, Jekyll Island.

Rossi, F., Van Beek, P., & Walsh, T. (2006). *Handbook of constraint programming (foundations of artificial intelligence)*. New York, NY: Elsevier Science Inc.

Rousseau, L. M., Gendreau, M., Pesant, G., & Focacci, F. (2004). Solving VRPTWs with constraint programming based column generation. *Annals of Operations Research*, *130*, 199. https://doi.org/10.1023/B:ANOR.0000032576.73681.29.

Salhi, S., Wassan, N., & Hajarat, M. (2013). The fleet size and mix vehicle routing problem with backhauls: Formulation and set partitioning-based heuristics. *Transportation Research Part E: Logistics and Transportation Review*, *56*, 22–35.

Schrijver, A. (1998). *Theory of linear and integer programming*. London: Wiley. ISBN 0- 471-98232-6.

SCIP. http://scip.zib.de/. Accessed 20 Jan 2016.

Sitek, P. (2014). A hybrid approach to the two-echelon capacitated vehicle routing problem (2E-CVRP). *Advances in Intelligent Systems and Computing*, *267*(2014), 251–263. https://doi.org/10.1007/978-3-319-05353-0_25.

Sitek, P., & Wikarek, J. (2015a). A hybrid programming framework for resource-constrained scheduling problems. In: *Intelligent data engineering and automated learning—IDEAL 2015, volume 9375 of the series lecture notes in computer science* (pp. 300–308).

Sitek, P., & Wikarek, J. (2015b). A hybrid framework for the modelling and optimisation of decision problems in sustainable supply chain management. *International Journal of Production Research*, *53*(21), 6611–6628. https://doi.org/10.1080/00207543.2015.1005762.

Sitek, P., & Wikarek, J. (2016). A hybrid programming framework for modeling and solving constraint satisfaction and optimization problems. *Scientific Programming*, *2016*, p. 13. Article ID 5102616. https://doi.org/10.1155/2016/5102616.

Sitek, P., Nielsen, I., & Wikarek, J. (2014). A hybrid multi-agent approach to the solving supply chain problems. *Procedia Computer Science*, *35*(KES), 1557–1566. https://doi.org/10.1016/j.procs.2014.08.239.

Toth, P., & Vigo, D. (2014) Vehicle routing: Problems, methods, and applications. In: SIAM, Philadelphia. https://doi.org/10.1137/1.9781611973594.fm.

Wassan, N., Wassan, N., Nagy, G., & Salhi, S. (2016). The multiple trip vehicle routing problem with backhauls: Formulation and a two-level variable neighbourhood search. *Computers and Operations Research*. https://doi.org/10.1016/j.cor.2015.12.017.

Yeun, L. C., Wan, R. I., Khairuddin, O., & Mourad, Z. (2008). Vehicle routing problem: Models and solutions. *Journal Of Quality Measurement And Analysis*, *4*(1), 205–218.

Zellner, M., Massey, D., Shiftan, Y., Levine, J., & Arquero, M. J. (2016). Overcoming the last-mile problem with transportation and land-use improvements: An agent-based approach. *International Journal of Transportation*, *4*(1), 1–26. https://doi.org/10.14257/ijt.2016.4.1.01.