CIRRELT

Centre interuniversitaire de recherche
sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

The Multi-Depot Inventory Routing Problem: An Application of Vendor-Management Inventory in City Logistics

Luca Bertazzi
Leandro C. Coelho
Annarita De Maio
Demetrio Laganà

August 2017

CIRRELT-2017-54

Document de travail également publié par la Faculté des sciences de l'administration de l'Université Laval, sous le numéro FSA-2017-011.

Bureaux de Montréal :
Université de Montréal
Pavillon André-Aisenstadt
C.P. 6128, succursale Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

Université Laval · McGill · Concordia University · ÉTS · UQÀM Université du Québec à Montréal · HEC Montréal · Polytechnique Montréal · Université de Montréal

The Multi-Depot Inventory Routing Problem: An Application of Vendor-Management Inventory in City Logistics

Luca Bertazzi[1], Leandro C. Coelho[2,*], Annarita De Maio[3], Demetrio Laganà[4]

[1] Department of Economics and Management, University of Brescia, Italy

[2] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Department of Operations and Decision Systems, 2325 de la Terrasse, Université Laval, Québec, Canada G1V 0A6

[3] Department of Mathematics and Computer Science, University of Calabria, Italy

[4] Department of Mechanical, Energy and Management Engineering-DIMEG, University of Calabria, Italy

Abstract. The context of this paper lies on the application of the Vendor-Managed Inventory paradigm in city-logistics. We formally model and solve the problem as a Multi-Depot Inventory Routing Problem (MDIRP). In general, the MDIRP is an NP-hard problem that aims at optimizing the trade-off between inventory and transportation management in an integrated way. With respect to the state of the art, a different context is presented, characterized by a complex urban environment. We formulate the problem, we design a branch--and--cut algorithm to solve it and then we propose a three--phase matheuristic to solve the problem efficiently. The urban space is partitioned into clusters allowing us to generate feasible routes for the MDIRP. In the first phase of the matheuristic, an integer program is solved to build clusters, while the second phase generates routes. Finally, in the third phase, a route-based formulation of the problem is solved to provide a feasible MDIRP solution. More emphasis is devoted to simultaneously balancing several factors that impact the clustering and route construction phases: distances, demand and inventory levels, time horizon extension and vehicle capacity. Computational experiments show that the matheuristic is very effective.

Keywords. Logistics, multi-depot inventory routing problem, city-logistics, mixed-integer linear programming, matheuristic.

Acknowledgements. The work by Leandro C. Coelho was partly supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grant 2014-05764. This support is gratefully acknowledged.

Results and views expressed in this publication are the sole responsibility of the authors and do not necessarily reflect those of CIRRELT.

Les résultats et opinions contenus dans cette publication ne reflètent pas nécessairement la position du CIRRELT et n'engagent pas sa responsabilité.

---

* Corresponding author: Leandro.Coelho@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2017

© Bertazzi, Coelho, De Maio, Laganà and CIRRELT, 2017

1 Introduction

In the last decades companies increased their interest in optimizing their supply chains. The spread of globalization and the development of information and communication technologies stimulated research towards the development of integrated logistics models with the aim of improving coordination. The main contributions in this direction are devoted to develop optimization models focused on two or more sequential logistics activities in the supply chain, such as inventory and routing, production or inventory, location and routing. All the resulting optimization models are based on two–echelon networks, in which one or more suppliers provide freight to many retailers or customers. Our problem falls within the field of two–echelon optimization problems in which customers must be supplied from different depots over a finite planning horizon, while transportation and inventory costs are minimized.

In this setting, *Vendor-Managed Inventory* (VMI) gained importance in different companies. VMI consists of a vendor/supplier, and a set of customers/retailers located in a given geographical area. The supplier monitors the inventory and decides on the replenishment policy of each retailer. The VMI setting assigns to the supplier the role of leading actor in the decisional process, in order to establish when and how much to deliver. This system applies a win-win strategy, because it guarantees an overall reduction of the logistic cost for the supplier and saving in the ordering cost for the customers. The development of the VMI setting derives from successful industrial applications in different fields: Procter & Gamble and Walmart (Barratt and Oliveira [2001]), chemical products (Dauzère-Pérès et al. [2007]), oil and gas (Bell et al. [1983], Grønhaug et al. [2010], Shen et al. [2011]), fuel (Popović et al. [2012]) and maritime cement transportation (Christiansen et al. [2011]). For an in depth overview of VMI applications we refer to Andersson et al. [2010].

The optimization problem that integrates VMI with routing is the well–known *Inventory Routing Problem* (IRP). In comparison to the classical *Vehicle Routing Problem*

(VRP), the IRP shows an added complexity due to the integration of the inventory component into the multi-period decisional process. Usually, an IRP deals with minimizing the sum of inventory and routing costs during the planning horizon, while avoiding stock–outs at the customers. Three main decisions are relevant in an IRP: when and how much to deliver to customers and which routes to use. Uncertain IRP is not widely investigated due to the increased complexity of the problem (see Solyalı et al. [2012], Bertazzi et al. [2013], Coelho et al. [2014], Bertazzi et al. [2016]). Despite the fact that supply chains are usually characterized by a great level of uncertainty, the literature demonstrates that good results also derive from deterministic analysis. Deterministic IRPs are common both from a practical standpoint or as a research area. The single–product and single–vehicle IRP is studied in Bertazzi and Speranza [2002], Archetti et al. [2007], Solyalı and Süral [2008]. Several papers investigate the single–product and multi–vehicle IRP, including Coelho et al. [2012] and Adulyasak et al. [2013]. The multi–product and multi-vehicle IRP is studied in Coelho and Laporte [2013] and Cordeau et al. [2015]. Exact algorithms for the IRP include branch–and–cut, branch–and–price, and more recently, branch–price–and–cut (Desaulniers et al. [2015]).

Heuristic methods were successfully applied to large IRP instances. A basic heuristic approach consists of decomposing the IRP in sub–problems solved in hierarchical order. Recent years have seen a large use of hybrid heuristic algorithms, in which mathematical programming models are embedded into heuristic frameworks. These algorithms are denoted *matheuristics*, and provide good results in solving IRPs, as shown by Bertazzi and Speranza [2011] and Archetti et al. [2017]. A decomposition approach to tackle large scale instances of the IRP can be found in Campbell and Salvesbergh [2004], and Cordeau et al. [2015], while a matheuristic procedure is designed in Archetti et al. [2017]. For an in depth analysis of the state of the art, the reader can refer to the surveys and tutorials by Bertazzi and Speranza [2013] and Coelho et al. [2013].

City logistics refers to the management of urban freight transportation, based on inte-

grated logistic systems with consolidation and coordination, aimed at increasing efficiency and reducing environmental damage (see Crainic et al. [2009], Bektas et al. [2015], Koç et al. [2016], Savelsbergh and Van Woensel [2016]). In these systems, two main types of facilities are used: City Distribution Centers (CDC), where storage, sorting, consolidation activities are carried out, and Satellites, where transdock-type of operations are carried out, with no storage. A typical application is when logistics is managed in mega-cities. A mega–city can be defined as a metropolitan area with a total population in excess of ten million people. This enormous urban space is usually affected by social and infrastructural problems related to energy consuption, traffic congestion and air pollution. To face the high complexity of logistics in mega–cities, the main idea is to split the urban space into districts (clusters), each one with an independent inventory management of its stores. The best practice implemented by many companies (e.g., Procter & Gamble and Walmart) in mega–city areas is to locate a depot in each district to control the inventories of all the stores by using a VMI setting.

We study an IRP in which VMI is applied to the customers of a company located in a mega–city. Given a set of depots, we want to determine the vehicles assigned to each depot, the customers served by each depot on each day, when and how much to deliver to each customer, and which delivery routes to use. These depots are supplied on each day to face the orders they have to serve on that day. Inventory levels of the customers are managed by the supplier. However, the corresponding inventory cost is paid by the customers. Therefore, suppliers minimize their routing cost only, but guarantee that no stock–out occurs at the customers. This problem is a multi–depot and multi–vehicle IRP ($MDIRP$). The main difference of our problem with respect to the practice implemented in many companies is that the assignment of customers to depots is not fixed, hence we are more flexible.

In this paper we define, model and solve a multi–vehicle and multi–depot IRP by proposing an effective three-phase matheuristic algorithm. In the first phase, an integer

program is used to build clusters on the basis of a quantitative measure of customers' critical level. In the second phase, a set of intra and inter-cluster routes is generated by considering the limited amount of resources in terms of vehicle capacity and maximum number of customers that can be served from a supplier. In the third phase, a route–based mixed-integer linear programming ($MILP$) formulation is solved to obtain a feasible solution for the problem. The results are compared with the ones obtained by solving a $MILP$ formulation using a commercial solver, through a branch–and–cut algorithm. The proposed matheuristic largely outperforms the branch–and–cut in terms of solution quality and computational time.

The remainder of the paper is organized as follows. The problem is formally described in Section 2. In Section 3 a mathematical formulation of the multi–depot and multi–vehicle IRP is presented. A branch–and–cut algorithm to solve the problem is described in Section 4. A matheuristic algorithm is presented in Section 5. In Section 6 extensive computational results are presented and discussed. Finally, conclusions are drawn in Section 7.

2 Problem description

We consider a complete undirect graph $G = (V, E)$, where V is the set of vertices and E is the set of edges. We partition the set V in such a way that $V = P \cup I$, where $P = \{1, 2, \ldots, m\}$ is the set of depots that deliver a product to the set of customers $I = \{m+1, m+2, \ldots, m+n\}$ over a finite and discrete time horizon H. Let $T = \{1, 2, \ldots, H\}$ be the corresponding set of time periods. A non–negative cost c_{ij} is associated with each edge $(i, j) \in E$. We assume that G is an Euclidean graph, so the triangular inequality holds. A set $K = \{1, 2, \ldots, M\}$ of vehicles, with the same capacity C, is available to perform deliveries. Each vehicle $k \in K$ can be assigned to one depot $p \in P$ for each time period $t \in T$. Each customer $i \in I$ can be served by different vehicles in the same time

period $t \in T$, either assigned to the same or different depots. A maximum inventory level U_i and a given starting inventory level I_{i0} are associated with each customer $i \in I$. We assume that both U_i and I_{i0} are integer. In each time period t, a deterministic integer demand d_{it} of each customer i must be satisfied. The inventory level of each customer i cannot be negative in each time period $t \in T \cup \{H+1\}$, i.e., stock–out is not allowed. The aim is to determine:

1. the set of vehicles to assign to each depot at each time period,
2. the quantity of product to deliver to each customer at each time period by using each vehicle,
3. the set of routes to travel at each time period,

that minimize the total transportation cost over the time horizon.

3 Mathematical formulation

We now present a mathematical formulation of the problem, which is adapted from Archetti et al. [2007], in which the multi–depot multi-vehicle case is tackled by using binary variables. We introduce the following notation. Let $\delta(S)$ be the set of edges (i, i') incident to the vertices $i \in S \subset V$ (edge cutset); for the sake of notation, if $S = \{i\}$, we denote the corresponding edge cutset as $\delta(i)$. Let $E(U)$ be the set of edges (i, j) such that $i, j \in U$, where $U \subseteq I$ is a given set of customers. Our mathematical formulation is based on the following variables:

- I_{it}: inventory level at customer i at the beginning of time period t;
- y_{iktp}: quantity to deliver to customer i in time period t by vehicle k starting the route from depot p;

- x_{ijktp}: binary variable equal to 1 if vehicle k starting from depot p travels directly from vertex i to vertex j in time period t, and equal to 0 otherwise;
- z_{iktp}: binary variable equal to 1 if vehicle k from depot p visits customer i in time period t, and equal to 0 otherwise;
- z_{pktp}: binary variable equal to 1 if vehicle k, assigned to depot p, starts its route from depot p in time period t, and equal to 0 otherwise.

The mathematical formulation is described by (1)–(14).

$$\min \quad \sum_{t\in T}\sum_{(i,j)\in E}\sum_{k\in K}\sum_{p\in P} c_{ij}x_{ijktp} \tag{1}$$

$$\text{s.t.} \quad I_{it} = I_{it-1} + \sum_{k\in K}\sum_{p\in P} y_{ikt-1p} - d_{it-1} \qquad \forall t \in T\cup\{H+1\}, \forall i \in I \tag{2}$$

$$I_{it} + \sum_{p\in P}\sum_{k\in K} y_{iktp} \leq U_i \qquad \forall t \in T, \forall i \in I \tag{3}$$

$$\sum_{i\in I} y_{iktp} \leq C\ z_{pktp} \qquad \forall t \in T, \forall p \in P, \forall k \in K \tag{4}$$

$$\sum_{i\in I} y_{iktp} \geq z_{pktp} \qquad \forall t \in T, \forall p \in P, \forall k \in K \tag{5}$$

$$y_{iktp} \leq C\ z_{iktp} \qquad \forall i \in I, \forall t \in T, \forall p \in P, \forall k \in K \tag{6}$$

$$z_{bktp} = 0 \qquad \forall t \in T, \forall p, b \in P, p \neq b, \forall k \in K, \tag{7}$$

$$\sum_{p\in P} z_{pktp} \leq 1 \qquad \forall t \in T, \forall k \in K \tag{8}$$

$$\sum_{j\in I:(j,i)\in E} x_{jiktp} + \sum_{j\in I:(i,j)\in E} x_{ijktp} = 2z_{iktp} \qquad \forall i \in I, \forall t \in T, \forall p \in P, \forall k \in K \tag{9}$$

$$\sum_{(i,j)\in E(S)} x_{ijktp} \leq \sum_{i\in S} z_{iktp} - z_{uktp} \qquad \forall S \subseteq I, |S| \geq 2, \forall t \in T, \forall k \in K, \forall p \in P, u \in S \quad (10)$$

$$x_{ijktp}, x_{pjktp}, x_{jpktp} \in \{0,1\} \qquad \forall i,j \in I, \forall t \in T, \forall p \in P, \forall k \in K \quad (11)$$

$$I_{it} \geq 0 \qquad \forall i \in I, \forall t \in T \cup \{H+1\} \quad (12)$$

$$y_{iktp} \geq 0 \qquad \forall i \in I, \forall t \in T, \forall p \in P, \forall k \in K \quad (13)$$

$$z_{iktp} \in \{0,1\} \qquad \forall i \in I, \forall t \in T, \forall p \in P, \forall k \in K. \quad (14)$$

The objective function (1) minimizes the total transportation cost. Constraints (2) balance the inventory flow at the customers. Note that the variable I_{i1} is equal to the initial inventory level I_{i0}. Constraints (3) guarantee that the maximum inventory level of each customer does not exceed U_i. Constraints (4) ensure that the total quantity loaded on vehicle k departing from depot p in period t does not exceed the capacity C, and, together with constraints (5) guarantee that, if the total quantity is greater than 0, then the corresponding vehicle is used. Constraints (6) enforce that, if a positive quantity of product is delivered to customer i with vehicle k starting from depot p in time period t, then customer i is visited. Constraints (7) guarantee that a vehicle k assigned to depot b in time period t cannot start a route from a different depot p in the same time period. Constraints (8) impose that each vehicle k can be assigned to at most one depot p in each time period t. Constraints (9) and (10) control the degree of the vertices and prohibit subtours, respectively. Finally, constraints (9)–(14) define the integrality and non-negativity conditions for the variables. Note that constraints (12) also guarantee that no stock-out occurs at each customer $i \in I$.

4 Branch–and–cut algorithm

We design a basic branch–and–cut algorithm to solve the problem. The initial linear programming relaxation (LP) is obtained by removing constraints (10) from the problem formulation (1)–(14), adding the following constraints that guarantee that each customer

i is visited at most once from vehicle k in each time period t:

$$\sum_{p \in P} z_{iktp} \leq 1 \quad \forall t \in T, \forall k \in K, \forall i \in I, \tag{15}$$

and then by adding violated subtour elimination constraints for each period, for each vehicle and for each depot. Constraints (15) are redundant in model (1)–(14), but they can increase the value of the corresponding LP. Subtour elimination cuts are separated heuristically along the lines of the procedure designed by Ahr [2004]. More precisely, the heuristic finds all the connected components in the auxiliary graph induced by all the edges such that $\overline{x}_{ijktp} \geq \epsilon + \tau$, where $\overline{x}_{ijktp}$ is the value of variable x_{ijktp} on edge (i, j) in the current LP while $\epsilon \in \{0, 0.25, 0.50\}$ and τ is a tolerance. Whenever a subset of customers S_p disjoint from depot p is found, the corresponding subtour elimination constraint is added for $u = argmax_{i \in S_p} \{\overline{z}_{iktp}\}$, where $\overline{z}_{iktp}$ is the value of variable z_{iktp} in the current LP. This heuristic procedure is also applied to any integer solution of the relaxed formulation, so that the best integer solution of the branch–and–cut is connected to the depot. The branching rule to be used is determined by the MIP solver.

In order to improve the quality of the root node lower bound, the following valid inequalities are added to the LP described so far:

1. *Priority inequalities:*

$$z_{iktp} \leq z_{pktp}, \quad \forall i \in I, \forall k \in K, \forall p \in P, \forall t \in T. \tag{16}$$

   These valid inequalities were presented by Archetti et al. [2007] to improve the performance of the proposed branch-and-cut for the single–vehicle IRP.

2. *Logical inequalities:*

$$x_{ipktp} + x_{piktp} \leq 2\ z_{iktp}, \quad \forall i \in I, \forall t \in T, \forall p \in P, \forall k \in K \tag{17}$$

$$x_{ijktp} \leq z_{iktp}, \quad \forall i, j \in I, \forall t \in T, \forall p \in P, \forall k \in K. \tag{18}$$

These inequalities are inspired by the logical cuts of Fischetti et al. [1998] and Gendreau et al. [1998]. Inequalities (17) impose that if the depot p is the predecessor or the successor of customer i in the route executed in period t by vehicle k starting from depot p, then customer i has to be visited in period t by vehicle k starting from depot p. Inequalities (18) impose that if customer j is the successor of customer i in the route performed in period t by vehicle k starting from depot p, then customer i has to be visited in period t by vehicle k starting from depot p.

3. *Disaggregate parity inequalities:*

$$\sum_{(i,j)\in\delta(S)\setminus(F)} x_{ijktp} \geq \sum_{(i,j)\in(F)} x_{ijktp} - |F| + 1,$$
$$\forall t \in T, \forall p \in P, \forall k \in K, \forall F \subseteq \delta(S), |F| \text{ odd} \tag{19}$$

4. *Depot–Aggregate parity inequalities:*

$$\sum_{p\in P}\sum_{(i,j)\in\delta(S)\setminus(F)} x_{ijktp} \geq \sum_{p\in P}\sum_{(i,j)\in(F)} x_{ijktp} - |F| + 1,$$
$$\forall t \in T, \forall k \in K, \forall F \subseteq \delta(S), |F| \text{ odd} \tag{20}$$

These inequalities are added dynamically to the LP in each node of the branch–and–cut algorithm. Parity inequalities were introduced by Barahona and Grötschel [1986] as co–circuit inequalities. They are effective for problems with binary variables, in case that the parity of vertices is required. An example of application of these inequalities is presented by Padberg and Rinaldi [1990] for the Symmetric TSP polytope. Inequalities (19) and (20) are separated heuristically according to the procedure described by Aráoz et al. [2009]. Note that aggregate parity inequalities over all the vehicles are not valid, since split delivery is allowed.

An initial solution of the branch-and-cut is computed as follows. We solve a relaxed formulation of (1)–(14), named $RMDIRP$ in the sequel, in which the routing cost in the

objective function is replaced by $\sum_{t\in T}\sum_{k\in K}\sum_{p\in P}\sum_{i\in I} c_{pi} z_{iktp}$, where c_{pi} is the cost of the edge $(p,i) \in E$ connecting the depot p to the customer i. Moreover, all the routing constraints (9) and (10) and the corresponding variables x_{ijktp} are temporarily removed from the mathematical formulation. A set of feasible clusters (one for each vehicle) is defined on the basis of the values of variables z_{iktp} in any optimal $RMDIRP$ solution. For each cluster, the optimal TSP tour is computed and the corresponding value of the objective function (1) is then determined. The relaxed formulation is solved repeatedly by adding at every iteration the following diversification constraints, with the aim of obtaining different solutions:

1. Let $\overline{Z}_{ktp} = \{i \in I : \overline{z}_{iktp} = 1\}$ be the set of the customers served by vehicle k in period t from depot p in the current feasible $RMDIRP$ solution. A set of feasible routes is built by solving a TSP on the sub–graph induced by $\overline{Z}_{ktp} \cup \{p\}$, for each k, t and p.

2. The average routing cost is computed over the set of routes returned in step 1. All the routes with a cost greater than the average routing cost are considered as candidates for diversification.

3. For each candidate route, a set of moving nodes, B_{ktp}, is built as follows. The vertices served in the route are ordered according to their non–decreasing insertion cost. The insertion cost of the vertex i is defined as the difference between the optimal TSP tour cost to serve all the customers in the route and the optimal TSP tour cost to serve all the customers in the route except i. The first $\left\lfloor \frac{|\overline{Z}_{ktp}|}{2} \right\rfloor$ vertices are inserted in the set B_{ktp}.

4. The diversification constraint is formulated as follows:

$$\sum_{i\in B_{ktp}} (1 - z_{iktp}) + \sum_{i\in I\setminus B_{ktp}} z_{iktp} \geq \left\lceil \frac{|B_{ktp}|}{|P|} \right\rceil . \tag{21}$$

Observe that the number of customers that can be moved among the routes decreases with the number of depots. Inequalities (21) are added to the $RMDIRP$ and the new problem is re-optimized. For each problem with diversification constraints, a time limit of 20 minutes is imposed.

5. The procedure ends when a maximum number ω of iterations is reached or $GAP = \frac{(c_W - c_B)}{c_B} 100 \geq \vartheta$, where c_W and c_B are the costs of the worst and the best $RMDIRP$ solutions respectively, while ϑ is a gap limit.

At the end of the procedure the best solution found is used as starting solution for the branch–and–cut algorithm.

The pseudocode of this procedure is provided in Algorithm 1.

5 Matheuristic for the $MDIRP$

We propose a matheuristic algorithm for the solution of the $MDIRP$ able to solve realistic–size instances. It is based on the following three–phase decomposition approach:

- *Clustering phase*: an integer linear programming model is solved to generate a partition of the set of customers into a set of clusters, one cluster for each depot $p \in P$.
- *Routing construction phase*: a set $\mathcal{R}$ of routes is built for the clusters generated in the first phase. The routes are generated on the basis of several replenishment policies and assuming different vehicle capacities. Two types of routes are generated: *intra-cluster routes* and *inter-cluster routes*. In the first type, each route can visit only customers in the cluster, while in the second type each route can also visit borderline customers, i.e., customers not in the cluster, but close enough to it. This latter type of routes is introduced to add flexibility in the construction of the routes.
- *Optimization phase*: a binary linear programming model, referred to as *Route-based*

Algorithm 1 Initial solution for the branch-and-cut algorithm

Set $\kappa = 0$;

Let $\overline{s}_{\mathcal{RMDIRP}}$ be a solution for the $RMDIRP$

Set $Ws_{\mathcal{RMDIRP}} = Bs_{\mathcal{RMDIRP}} = \overline{s}_{\mathcal{RMDIRP}}$; where $Ws_{\mathcal{RMDIRP}}$ is the worst $RMDIRP$ solution, whose cost is c_W and $Bs_{\mathcal{RMDIRP}}$ is the best $RMDIRP$ solution whose cost is c_B;

Set $GAP = 0$, $k = 0$

while Problem is feasible and $GAP \leq \vartheta$ and $\kappa \leq \omega$ **do**

determine the route set $\mathcal{R}^{\kappa}$;

determine the average routing cost $c^{\kappa}_{\mathcal{R}}$ of routes in $\mathcal{R}^{\kappa}$

for each route $r \in \mathcal{R}^{\kappa}$ **do**

if $c_r \geq c^{\kappa}_{\mathcal{R}}$ **then**

build set B_{ktp}

add diversification constraint (21)

end if

end for

Solve the $RMDIRP$ with diversification constraints , and let $\overline{s}_{\mathcal{RMDIRP}}$ be the corresponding solution.

if $c_{s_{\mathcal{RMDIRP}}} < c_{Bs_{\mathcal{RMDIRP}}}$ **then**

Set $Bs_{\mathcal{RMDIRP}} = \overline{s}_{\mathcal{RMDIRP}}$

end if

if $c_{s_{\mathcal{RMDIRP}}} > c_{Ws_{\mathcal{RMDIRP}}}$ **then**

$Ws_{\mathcal{RMDIRP}} = \overline{s}_{\mathcal{RMDIRP}}$

end if

$GAP = \frac{(c_B - c_W)}{c_B} 100$

$\kappa = \kappa + 1$

end while

$MDIRP$, is optimally solved to obtain a feasible solution of the $MDIRP$, selecting routes from the set $\mathcal{R}$ and determining delivery quantities to each customer in a given period of the time horizon. This quantity can be different than the one used to generate the routes in the previous phase.

We now describe these three phases in detail.

5.1 Clustering phase

The aim of this phase is to partition the set of customers I into a set of clusters $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{|P|}\}$. We identify *critical customers* and build clusters having two main features: a limited number of customers and a maximum average critical level. The idea to identify critical customers in an IRP was proposed by Campbell and Salvesbergh [2004], where a qualitative definition of critical customers is proposed. Here, we provide a quantitative measure, computed on the basis of the minimum number of deliveries needed to serve the customer over the time horizon, and of the average distance of the customer from the depots. The corresponding critical level CL_i is computed as:

$$CL_i = \alpha R_i + (1 - \alpha) M_i, \tag{22}$$

where α takes values between 0 and 1, while R_i and M_i are respectively the minimum number of deliveries to customer i over the time horizon and the average delivery cost for each customer i with respect to all the depots. These parameters are obtained by computing:

$$\hat{R}_i = \frac{\max\left\{0, \sum_{t \in T} d_{it} - I_{i0}\right\}}{U_i}$$

$$\hat{M}_i = \frac{\sum_{p \in P} DC_i^p}{|P|},$$

where $DC_i^p = c_{TST}^p - c_{TST\setminus\{i\}}^p$ is an estimated transportation cost to serve customer i from depot p, computed as the difference between the optimal TSP tour cost to serve all customers in I from depot $p \in P$ (c_{TST}^p) and the optimal TSP tour cost to serve all customers $I\setminus\{i\}$ from depot $p \in P$ ($c_{TST\setminus\{i\}}^p$). The values of R_i and M_i in (22) are obtained by normalizing $\hat{R}_i$ and $\hat{M}_i$ in the interval $[0, 10]$, where 0 is assumed to be the minimum value and 10 to be the maximum value.

The values of CL_i are the input data of the following binary linear programming model aimed at building the clusters. Let CC be the maximum number of customers for each cluster, TC be the maximum value of the average critical level for each cluster computed with respect to CC, and b_{pi} a binary variable equal to 1 if customer i belongs to the cluster C_p, and 0 otherwise. Then, the model is formulated as follows:

$$\min \quad \sum_{(p,i)\in E} c_{pi}\ b_{pi} \tag{23}$$

$$\text{s.t.} \quad \sum_{i\in I} b_{pi} \leq CC \qquad \forall\ p \in P \tag{24}$$

$$\sum_{p\in P} b_{pi} = 1 \qquad \forall\ i \in I \tag{25}$$

$$\sum_{i\in I} CL_i b_{pi}/CC \leq TC \qquad \forall\ p \in P \tag{26}$$

$$b_{pi} \in \{0,1\} \qquad \forall\ p \in P,\ \forall i \in I. \tag{27}$$

The objective function (23) minimizes the total distance between customers and depots. Constraints (24) enforce the total number of customers in each cluster to be within the maximum number CC. Constraints (25) impose that each customer is assigned to exactly one cluster. Constraints (26) enforce the average critical level of each cluster to be not greater than the maximum value TC. Constraints (27) define variables b_{pi}.

In the following, we will refer this procedure as the one that receives an instance of the $MDIRP$ as input, I_{MDIRP}, and returns a set of clusters $\mathcal{C} = \{C_1, \ldots, C_{|P|}\}$, that is:

$\mathcal{C} \leftarrow Clustering\,(I_{MDIRP})$.

5.2 Routing construction phase

The aim of the second phase is to generate a restricted number of routes for the $MDIRP$ on the basis of the clusters generated by model (23)–(27), and on the basis of some replenishment policies. Two classes of routes are generated: *intra-cluster* and *inter-cluster routes*. The first class is composed of all routes serving customers that are in the same cluster, while the second class is composed of all the routes that serve also borderline customers. A similar idea is presented by Salhi et al. [2014] for a multi–depot multi–vehicle VRP. They divided customers in two sets: the first one composed of customers served by the depot nearest to them, and the second serving borderline customers. They define borderline customers as "the customers that happen to be situated approximately halfway between its nearest and its second nearest depots". They use this classification as starting point for routes generation. Here, we use a different clustering procedure, strongly focused on the nature of the $MDIRP$.

Let us first describe the intra-cluster routes we generate. For each cluster C_p, we focus on direct delivery routes from the depot to one customer and on routes built aggregating the customers served in the same time period based on the following replenishment policies:

- *Order–up–to level policy.* This policy aims at restoring the maximum inventory level at customer i. This policy is referred to as g_1. For each customer i in the cluster C_p and for each time period $t \in T$, the replenishment quantity $\hat{y}_{it}(g_1)$ is computed as follows:

$$\hat{y}_{it}(g_1) = \begin{cases} U_i - \hat{I}_{it-1}(g_1), & \text{if } d_{it} \geq \hat{I}_{it-1}(g_1) \\ 0, & \text{otherwise} \end{cases}$$

where $\hat{I}_{it-1}(g_1) = I_{i0} + \sum_{h=1}^{t-2} \hat{y}_{ih}(g_1) - \sum_{h=1}^{t-2} d_{ih}$.

- *Maximum level policy.* This policy aims to have an inventory level equal to the demand, whenever the current inventory level is lower than the demand. This policy is referred to as g_2. For each customer i in the cluster C_p and for each time period $t \in T$, the replenishment quantity $\hat{y}_{it}(g_2)$ is computed as follows:

$$\hat{y}_{it}(g_2) = \begin{cases} d_{it} - \hat{I}_{it-1}(g_2), & \text{if } d_{it} \geq \hat{I}_{it-1}(g_2) \\ 0, & \text{otherwise} \end{cases}$$

where $\hat{I}_{it-1}(g_2) = I_{i0} + \sum_{h=1}^{t-2} \hat{y}_{ih}(g_2) - \sum_{h=1}^{t-2} d_{ih}$.

- *Initial–inventory–level policy.* This replenishment policy aims at restoring the maximum between the initial inventory level I_{i0} and the demand d_{it} in each time period, whenever the current inventory level is lower than the demand. We refer to this policy as g_3. For each customer i in the cluster C_p and for each time period $t \in T$, the replenishment quantity $\hat{y}_{it}(g_3)$ is computed as follows:

$$\hat{y}_{it}(g_3) = \begin{cases} \max\left\{I_{i0} - \hat{I}_{it-1}(g_3), d_{it} - \hat{I}_{it-1}(g_3)\right\}, & \text{if } d_{it} \geq \hat{I}_{it-1}(g_3) \\ 0, & \text{otherwise} \end{cases}$$

where $\hat{I}_{it-1}(g_3) = I_{i0} + \sum_{h=1}^{t-2} \hat{y}_{ih}(g_3) - \sum_{h=1}^{t-2} d_{ih}$.

- *Critical–customers–level policy.* This replenishment policy aims at delivering the demand or twice its value, depending on the value of the critical level CL_i, to the customers not having enough inventory level to satisfy the demand. We refer to this policy as g_4. For each customer i in the cluster C_p and for each time period $t \in T$, the replenishment quantity $\hat{y}_{it}(g_4)$ is computed as follows:

$$\hat{y}_{it}(g_4) = \begin{cases} \min\left\{2d_{it}, U_i\right\}, & \text{if } d_{it} \geq \hat{I}_{it-1}(g_4), \text{ and } CL_i \geq 6 \\ d_{it}, & \text{if } d_{it} \geq \hat{I}_{it-1}(g_4), \text{and } CL_i < 6 \\ 0, & \text{otherwise} \end{cases}$$

where $\hat{I}_{it-1}(g_4) = I_{i0} + \sum_{h=1}^{t-2} \hat{y}_{ih}(g_4) - \sum_{h=1}^{t-2} d_{ih}$.

Note that the threshold on the critical level CL_i has been set equal to 6, that is an average value in the scale from 1 to 10.

Each replenishment policy allows us to define a set of customers served in each time period. If the vehicle capacity is respected, then a TSP route is generated starting from the depot and visiting all the served customers. If the capacity constraint is exceeded, some deliveries are moved to the previous or to the next day until feasibility is recovered. More precisely, the following steps are executed to construct a feasible delivery schedule:

- For each period $t = 1, \ldots, H-1$, if the vehicle capacity is exceeded, the customers to be served in t are sorted in non–decreasing value of the critical level CL_i and, following this ordering, they are moved to time period $t+1$ until the capacity constraint is satisfied. In $t+1$ all the replenishment quantities of the postponed customers are re-computed according to the replenishment policy.

- If the capacity constraint is violated in period $t = H$, customers are sorted in non–increasing value of the critical level CL_i and, following this ordering, they are moved to period $t-1$ until the capacity constraint is satisfied. In $t-1$ all the delivery quantities of the anticipated customers are re-computed according to the replenishment policy. This procedure is repeated until $t = 2$.

- For each time period, a delivery route is built by solving a TSP on the sub–graph induced by all the customers to be served in that time period and the corresponding depot.

This generation of intra-cluster routes for cluster C_p is executed with three different values of the capacity Cap of the vehicle: C, $C/2$ and $C/3$. This meets the aim of generating different routes. In the following, we will refer to this procedure as Intra–cluster Route Generation ($INTRARG$), which operates on the basis of a given time horizon H, replenishment

policy g, cluster C_p and capacity Cap, and returns a set of TSP routes over H supplying customers of C_p with policy g, $TSPR^g_{C_p}$, that is: $TSPR^g_{C_p} \leftarrow INTRARG\,(H, g, C_p, Cap)$.

Let us now describe how inter-cluster routes are generated. For each cluster C_p:

- Build the corresponding rectangular convex hull, i.e., the smallest rectangular area including all the points associated with the geographical coordinates of the customers in the cluster. Let X^p_{min}, X^p_{max}, Y^p_{min} and Y^p_{max} be the minimum and the maximum values of customers' coordinates, respectively, of the cluster C_p. The corresponding rectangular convex hull is built over the following vertices: (X^p_{min}, Y^p_{min}), (X^p_{min}, Y^p_{max}), (X^p_{max}, Y^p_{min}) and (X^p_{max}, Y^p_{max}). Let DG_p be the diagonal of this rectangular convex hull (see Figure 1).

- Build the set $Bord_p$ of the borderline customers: this set is composed of all the customers not included in C_p whose distance from the nearest extreme vertex of the rectangular convex hull is less than λDG_p, where $0 < \lambda < 1$ is a parameter set on the basis of the number of clusters.

- Build the inter-cluster routes serving the customers in the set $Bord_p \cup C_p$: they are built using the same procedure adopted to generate intra–cluster routes, in which only g_1, g_2 and g_3 and capacity Cap equal to C, $C/2$ and $C/3$ are considered (see Figure 1).

In the following, we will refer to this procedure as Inter–cluster Route Generation ($INTERRG$), which operates on the basis of a given time horizon H, replenishment policy g, cluster $Bord_p \cup C_p$ and of capacity Cap, and returns a set of TSP routes over H supplying customers of $Bord_p \cup C_p$ with policy g, $TSPR^g_{Bord_p \cup C_p}$, that is: $TSPR^g_{Bord_p \cup C_p} \leftarrow INTERRG\,(H, g, Bord_p \cup C_p, Cap)$.

A set $\mathcal{R}$ of delivery routes is built from the union of the set of all the intra–cluster routes and of the set of all the inter–cluster routes, excluding all duplicated routes.

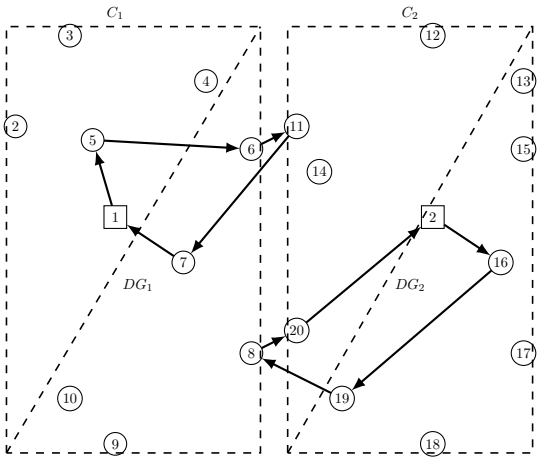

Figure 1: Examples of rectangular convex hull and inter-cluster routes.

5.3 Optimization phase

In the third phase, the following route-based $MDIRP$, based on the set of routes $\mathcal{R}$, is solved to determine a feasible solution of the $MDIRP$. We introduce the following parameters that use an explicit index of the routes $r \in \mathcal{R}$:

- c_r: cost of route $r \in \mathcal{R}$
- a_{ir}: binary parameter equal to 1 if customer i is served in route r, 0 otherwise

and the following decision variables:

- m_{irt}: quantity shipped to customer i in route r in time period t
- Inv_{it}: inventory level of customer i at the beginning of time period $t \in T \cup \{H+1\}$
- n_{rt}: binary variable equal to 1 if route r is used in time period t and 0 otherwise.

The problem is then formulated as follows:

$$\min \quad \sum_{t\in T}\sum_{r\in\mathcal{R}} c_r\ n_{rt} \tag{28}$$

$$\text{s.t.} \quad Inv_{i,t} = Inv_{it-1} + \sum_{r\in R} m_{irt-1} - d_{it-1}, \qquad \forall i \in I, \forall t \in T \cup \{H+1\} \tag{29}$$

$$\sum_{r\in\mathcal{R}} m_{irt} + I_{it} \leq U_i, \qquad \forall t \in T, \forall i \in I \tag{30}$$

$$\sum_{i\in I} m_{irt} \leq C\ n_{rt}, \qquad \forall r \in \mathcal{R}, \forall t \in T \tag{31}$$

$$m_{irt} \leq C\ a_{ir}, \qquad \forall i \in I, \forall r \in \mathcal{R}, \forall t \in T \tag{32}$$

$$\sum_{r\in\mathcal{R}} n_{rt} \leq M, \qquad \forall t \in T \tag{33}$$

$$Inv_{it} \geq 0, \qquad \forall i \in I, \forall t \in T \cup \{H+1\} \tag{34}$$

$$m_{irt} \geq 0, \qquad \forall i \in I, \forall r \in \mathcal{R}, \forall t \in T \tag{35}$$

$$n_{rt} \in \{0,1\}, \qquad \forall r \in \mathcal{R}, \forall t \in T. \tag{36}$$

The objective function (28) minimizes the total routing cost. Constraints (29) define the inventory level at each customer i at each time period t. Constraints (30) enforce the maximum inventory level of each customer i at each time period t to be not greater than U_i. Constraints (31) enforce the total amount delivered with each route r in time period t to be within the vehicle capacity, while constraints (32) guarantee that a quantity can be delivered to customer i by vehicle r in period t only if customer i is served by route r. Constraints (33) enforce the number of routes used in each time period t to be within the fleet size M. Finally, constraints (34)–(36) define the decision variables.

The overall scheme of the matheuristic algorithm designed for the $MDIRP$ is described in Algorithm 2.

Algorithm 2 Matheuristic for the $MDIRP$

$\mathcal{R} := \emptyset$

Phase 1: clustering

for $i \in I$ **do**

compute parameter R_i, M_i, $CL_i = \alpha R_i + (1-\alpha)M_i$

end for

solve the **clustering** problem (23)–(27), $\mathcal{C} \leftarrow Clustering\,(I_{MDIRP})$

Phase 2: Routing construction

for each cluster $C_p \in \mathcal{C}$ **do**

for each customer $i \in C_p$ **do**

determine the direct delivery routes r_i^p from depot $p \in C_p$ to i

add this route to $\mathcal{R}$: $\mathcal{R} := \mathcal{R} \cup \{r_i^p\}$

end for

for each replenishment policy g_1, g_2, g_3, g_4 **do**

for each capacity $Cap \in \left\{Q, \frac{Q}{2}, \frac{Q}{3}\right\}$ **do**

for each customer $i \in C_p$ **do**

for each $t \in T$ **do**

compute the replenishment quantity $\hat{y}_{it}(g)$;

end for

end for

determine $TSPR^g_{C_p} \leftarrow INTRARG\,(H, g, C_p, Cap)$

update $\mathcal{R}$: $\mathcal{R} = \mathcal{R} \cup TSPR^g_{C_p}$

determine $TSPR^g_{Bord_p \cup C_p} \leftarrow INTERRG\,(H, g, Bord_p \cup C_p, Cap)$

update $\mathcal{R}$: $\mathcal{R} = \mathcal{R} \cup TSPR^g_{Bord_p \cup C_p}$

end for

end for

end for

Phase 3: Optimization

eliminate from $\mathcal{R}$ all the duplicated routes

solve the **route-based** $MDIRP$ formulation (28)–(36)

6 Computational results

The branch-and-cut algorithm described in Section 4 and the matheuristic described in Section 5 were coded in C++ and compiled with g++ -O3. Computational experiments were carried out on a PC equipped with an Intel Core i7-6500U CPU running at 2.50 GHz, with 8 GB of RAM with the scientific Linux 6.6 operating system. We use the MIP solver IBM CPLEX 12.6.1 using its default settings. To solve the TSP we use the *Concorde TSP Solver*.

The performance of the algorithms is evaluated on a set of instances derived from the benchmark set of Archetti et al. [2007]. We maintain the original depot as the first depot and we generate the remaining $p-1$ depots randomly. Our data set is composed of 100 instances with 5 to 50 customers. For each number of customers, a group of five instances is generated with a number of depots increasing from 2 to 6 according to the size of the instance.

The time horizon H is 3 and 6. In order to generate multi–vehicle instances, we consider a fleet of 3 vehicles with a capacity that is reduced by $\frac{1}{3}$ up to 1, with respect to the capacity in the original instances. Instances are labelled as *SnNdDhH*, where S indicates the instance number, N is the number of customers in the instance, D is the number of depots, and H is the number of periods. A time limit of 3 hours was imposed to CPLEX for solving the mathematical model given by (28)–(36), while the branch-and-cut was run with a time limit of 6 hours. The following parameters are set after preliminary tests on a subset of instances: $\epsilon = 0.2$, $\omega = 10$, $\alpha = 0.2$, $TC = 6$, $CC \in \{3, \ldots, 15\}$ and $\lambda \in \{0.2, \ldots, 0.5\}$.

6.1 Branch-and-cut performance

Tables 1 and 2 provide results for the branch–and–cut algorithm in the instances with $H = 3$ and $H = 6$, respectively. The tables are organized as follows. Column **Instance** provides the instance name. Column **MIPstart** provides the value of the initial feasible solution. Column **Best LB** provides the value of the best lower bound returned by the branch–and–cut at the end of the time limit, column **Cost** reports the cost of the best solution found by the branch–and–cut algorithm, column **Time (s)** shows the computational time in seconds, while column **Nodes** provides the number of nodes processed in the branch-and-cut algorithm. Columns **Subtours**, **Dis Parity Ineqs** and **Aggr Parity Ineqs** report the number of added sub–tour elimination constraints, disaggregate parity and aggregate parity inequalities that are added dynamically to the current LP , in any node of the branch–and–cut tree, respectively. Finally, column **Gap (%)** provides the optimatility gap. These results show that, even if an initial solution is provided and several families of cuts are used, the problem remains very challenging to be solved exactly. Even for small instances with 10 customers and 3 periods, optimality is not achieved for all instances. Several instances with more than 35 customers used all the computational time at the root node only, and the average gap was 31.24% for instances with 3 periods and 31.59% for those with 6 periods.

6.2 Matheuristic algorithm performance

Tables 3 and 4 show the results provided by the matheuristic algorithm. These tables are organized as follows. Column **Instance** denotes the instance name. Column **N. Clust.** reports the number of generated clusters, while column $|C_i|$ reports the cardinality of each cluster. Column **N. Routes** provides the number of intra–cluster and inter–cluster routes. The computational time of the algorithm, expressed in seconds, is reported in column **Time (s)**, while the cost of the best solution is reported in column **Cost**. Finally,

Instance	MIPstart	Best LB	Cost	Time (s)	Nodes	Subtours	Dis Par. Ineqs	Aggr Par. Ineqs	GAP %
1n5d2h3	1148.80	1148.80	1148.80	89.71	500	99	209	118	0.00
2n5d2h3	1043.20	956.24	956.24	434.76	1041	112	165	155	0.00
3n5d2h3	2294.50	1801.02	1801.02	360.51	535	163	374	287	0.00
4n5d2h3	1486.91	142.39	1425.39	1537.42	11851	75	193	134	0.00
5n5d2h3	1808,40	1808.40	1808.40	60.09	260	87	162	98	0.00
1n10d2h3	2177.98	1980,2831	2177,9892	t.l.	6621	917	741	553	10.91
2n10d2h3	2916.24	2376.56	2427.66	21700	7749	897	699	500	2.11
3n10d2h3	1651.54	1651.54	1651.54	1854.78	1609	424	365	218	0.00
4n10d2h3	2650.52	2449.3	2449.3	1994.27	473	285	287	157	0.00
5n10d2h3	1982.90	1982.90	1982.90	734	428	305	319	190	0.00
1n15d2h3	4588.88	3333.99	4588.88	21700	2022	1540	1110	721	27.35
2n15d2h3	2637.48	2436.43	2436.43	20955	5707	841	773	550	0.00
3n15d2h3	3826.82	3075.69	3075.69	6602.50	1846	853	753	442	0.00
4n15d2h3	2708.29	1986.95	2481.13	21700	1951	1654	1071	753	20.00
5n15d2h3	3321.56	2106.66	2912.07	21700	1923	1659	1046	675	37.69
1n20d3h3	3925.25	2632.99	3263.25	21700	834	1890	1462	678	19.32
2n20d3h3	5077.94	3487.4	4395.27	21700	276	1277	1562	812	20.66
3n20d3h3	4423.09	2916.28	3399.42	21700	808	1808	1673	818	14.21
4n20d3h3	4779.5359	3565.98	4501.08	21700	141	1117	1272	632	20.78
5n20d3h3	4848.89	3303.17	4338.40	21700	290	1141	1516	660	23.86
1n25d4h3	4203.45	2861.09	4203.45	21700	51	1187	1790	815	31.93
2n25d4h3	4570.13	2942.65	4570.13	21700	149	1721	1871	796	35.61
3n25d4h3	4537.87	2746.15	4127.82	21700	155	1619	1411	655	33.47
4n25d4h3	5722.98	3311.13	5722.99	21700	28	1374	2128	824	42.14
5n25d4h3	6214.81	2686.02	6214.81	21700	46	1253	1137	460	56.78
1n30d4h3	5206.49	3466.25	5206.49	21700	17	1270	1757	761	33.42
2n30d4h3	4805.12	3078.97	4805.12	21700	44	1583	2484	1023	35.92
3n30d4h3	5344.53	3674.43	5344.53	21700	11	712	1762	733	31.25
4n30d4h3	4805.12	2748.57	4805.12	21700	23	1040	1015	429	42.80
5n30d4h3	4693.64	2906.89	4693.64	21700	49	1366	1663	791	38.07
1n35d5h3	7013.75	4282.38	7013.75	21700	0	530	2477	798	38.94
2n35d5h3	6138.55	3362.17	6138.55	21700	12	1368	1446	618	45.23
3n35d5h3	6228.25	2977.10	6228.25	21700	2	872	768	250	52.20
4n35d5h3	6660.30	3481	6660.3	21700	5	733	769	237	47.73
5n35d5h3	5771.25	3078.37	5771.25	21700	3	1094	946	305	46.66
1n40d5h3	7080.27	3561.34	7080.27	21700	0	534	2189	625	49.70
2n40d5h3	7128.72	2940.72	7128.72	21700	3	1664	1233	575	58.75
3n40d5h3	6434.14	3394.62	6434.14	21700	3	1187	1932	882	47.24
4n40d5h3	5742.3	3357.24	5742.3	21700	3	1169	1473	743	41.54
5n40d5h3	5191.53	1938.53	5191.53	21700	2	1369	1766	707	62.70
1n45d6h3	7272.02	3922.67	7272.02	21700	0	502	3171	1483	46.06
2n45d6h3	6634.67	3677.20	6634.67	21700	1	1196	1402	705	44.58
3n45d6h3	6721.43	3267.56	6721.43	21700	1	1028	1243	610	51.39
4n45d6h3	6292.32	4091.47	6292.32	21700	1	1100	943	531	34.98
5n45d6h3	5650.86	2974.26	5650.86	21700	0	531	620	352	47.37
1n50d6h3	7881.37	3762.88	7881.37	21700	0	509	1957	918	52.26
2n50d6h3	7432.51	3293.44	7432.51	21700	1	672	1920	818	55.69
3n50d6h3	7611.76	3570.26	7611.76	21700	0	530	864	512	53.10
4n50d6h3	7740.36	3458.57	7740.36	21700	0	545	832	506	55.32
5n50d6h3	6509.11	3095.87	6509.11	21700	0	508	1348	627	52.44
Average				18052.46					31.24

Table 1: Branch–and–cut performance for instances with $H = 3$

Instance	MIPstart	Best LB	Cost	Time (s)	Nodes	Subtours	Dis Par. Ineqs	Aggr Par. Ineqs	Gap %
1n5d2h6	2804.46	2595.14	2595.14	8685.16	13970	370	691	344	0.00
2n5d2h6	3296.32	2841.92	3296.32	21700	10034	339	811	534	13.79
3n5d2h6	5287.78	4488.81	5145.63	21700	33006	474	654	522	12.76
4n5d2h6	2963.12	2604.27	2963.12	21700	10975	305	486	394	12.11
5n5d2h6	3120.91	2717.77	3120.91	21700	8894	514	937	639	12.92
1n10d2h6	5102.45	3237.99	5102.45	21700	1871	1304	1456	756	36.54
2n10d2h6	6071.31	3968.77	5501.12	21700	1823	1344	1462	657	27.86
3n10d2h6	5964.91	3755.90	5617.17	21700	2212	1450	1778	914	33.14
4n10d2h6	6345.62	4301.26	5849.66	21700	2310	1654	1377	723	26.48
5n10d2h6	5651.00	3708.58	5077.5	21700	2283	1395	1312	666	26.96
1n15d2h6	9904.03	6772.05	9904.03	21700	599	1833	2232	917	31.62
2n15d2h6	10202.70	6862.56	10202.70	21700	377	1563	1753	764	32.74
3n15d2h6	9276.72	5629.02	9276.72	21700	1205	1845	1142	571	39.34
4n15d2h6	9717.60	6376.08	8875.49	21700	732	1717	1230	631	28.16
5n15d2h6	10439.87	6710.09	8997.26	21700	709	1679	1566	707	25.42
1n20d3h6	11433.4	11433.40	7273.57	21700	22	1171	1284	2286	36.44
2n20d3h6	9094.27	5466.73	9094.27	21700	64	1343	1970	858	39.89
3n20d3h6	10807.75	7825.83	10807.75	21700	34	1604	1967	847	27.59
4n20d3h6	12965.1	6365.89	12965.1	21700	46	1771	2749	1224	50.90
5n20d3h6	13545.80	6736.51	13545.80	21700	34	1592	2497	1174	52.93
1n25d4h6	11949.99	6751.13	11949.99	21700	10	1915	1127	598	31.62
2n25d4h6	10369.70	5901.98	10369.70	21700	3	1383	1178	619	43.08
3n25d4h6	10511.30	6293.80	10511.30	21700	1	1205	2842	1287	40.12
4n25d4h6	10457.1	5685.74	10457.1	21700	1	1145	1080	485	45.63
5n25d4h6	9556.95	5530.76	9556.95	21700	1	1093	1085	472	42.13
1n30d4h6	12757.90	6397.71	12757.90	21700	1	1160	1464	773	49.85
2n30d4h6	12848.94	b7117.92	12848.94	21700	1	1282	1572	732	44.60
3n30d4h6	11921.20	6454.06	11921.20	21700	0	515	2506	1130	45.86
4n30d4h6	11420.40	6633.43	11420.40	21700	1	1248	1512	692	41.92
5n30d4h6	9960.2	6499.14	9960.2	21700	1	784	2558	1228	34.75
1n35d5h6	12924	7768.89	12924	21700	0	424	361	243	39.89
2n35d5h6	11561.1	6631.83	11561.1	21700	1	1199	1102	641	42.64
3n35d5h6	11676.10	6923.28	11676.10	21700	1	984	1176	698	40.71
4n35d5h6	11004.45	7015.47	11004.45	21700	0	555	924	673	36.25
5n35d5h6	15450.90	8937.02	15450.90	21700	0	513	1115	806	42.16
1n40d5h6	12802.5	7350.23	12208.5	21700	1	803	950	624	42.59
2n40d5h6	13999.70	9062.14	13999.70	21700	0	518	1756	1086	35.27
3n40d5h6	10887.50	6144.44	10887.50	21700	1	1701	1253	744	43.84
4n40d5h6	12554.70	6584.41	12554.70	21700	0	440	325	184	43.02
5n40d5h6	13283.7	797.05	13283.7	21700	0	458	499	341	39.96
1n45d6h6	10884.9	6872.81	10884.9	21700	0	505	466	271	36.86
2n45d6h6	11596.40	6837.06	11596.40	21700	0	552	1038	774	41.04
3n45d6h6	11639.20	6896.55	11639.20	21700	0	625	774	950	40.75
4n45d6h6	12116.65	6608.97	12116.65	21700	0	575	440	362	45.65
5n45d6h6	13737.20	7627.50	13373.20	21700	0	595	584	348	44.48
1n50d6h6	14031.9	7288.71	14031.9	21700	0	159	144	120	48.06
2n50d6h6	14635.32	6775.59	14635.32	21700	0	605	590	470	41.78
3n50d6h6	13721.20	7205.80	13721.20	21700	0	160	175	143	47.48
4n50d6h6	13202.43	7332.91	13202.43	21700	0	115	123	106	44.46
5n50d6h6	13676.2	6805.53	13676.2	21700	0	174	173	114	50.24
Average				21439.71					36.22

Table 2: Branch-and-cut performance for instances with H= 6

column **MHIRP GAP%** shows the percentage gap provided by CPLEX to solve model (28)–(36) within the time limit of 3 hours. Unlike the exact method, our matheuristic provided solutions within a few minutes of computing time. The results show that clusters are balanced in terms of number of customers and that the number of generated routes is very small with respect to the possible number of routes. The average number of routes that are generated to 462 for the set with $H = 3$, and is equal to 492 for the set with $H = 6$. The number of routes exceeds 1000 in only 6% of the instances, the ones with the largest number of customers. The average computational time is 88 seconds for the data set with $H = 3$, while is around 1965 seconds for the data set with $H = 6$.

Instance	N.Clust.	$\|C_i\|$	N. Routes	Time (s)	Cost	MHIRP GAP %
1n5d2h3	2	4,3	10	0.09	1148.80	0.00
2n5d2h3	2	4,3	16	0.1	956.24	0.00
3n5d2h3	2	3,4	13	0.13	1801.02	0.00
4n5d2h3	2	3,4	12	0.12	1425.39	0.00
5n5d2h3	2	3,4	10	0.08	1808.4	0.00
1n10d2h3	2	5,7	71	1.76	2112.53	0.00
2n10d2h3	2	5,7	75	2.21	2425.97	0.00
3n10d2h3	2	8,4	65	0.83	1651.54	0.00
4n10d2h3	2	7,5	59	0.81	2449.30	0.00
5n10d2h3	2	7,5	62	0.89	1982.90	0.00
1n15d2h3	2	7,10	188	3.76	3891.56	0.00
2n15d2h3	2	7,10	169	4.3	2436.43	0.00
3n15d2h3	2	9,8	176	3.84	3189.5	0.00
4n15d2h3	2	8,9	180	3.88	2298.16	0.00
5n15d2h3	2	6,11	222	1.93	2329.42	0.00
1n20d3h3	3	7,11,5	255	11.66	3095.91	0.00
2n20d3h3	3	7,11,5	255	6.44	4074.56	0.00
3n20d3h3	3	7,11,5	269	5.79	3361.74	0.00
4n20d3h3	3	8,10,5	252	7.88	4151.87	0.00
5n20d3h3	3	5,11,7	273	27.81	4235.63	0.00
1n25d4h3	4	6,10,5,8	277	13.97	3354.83	0.00
2n25d4h3	4	9,7,5,8	263	11.07	3654.25	0.00
3n25d4h3	4	9,8,6,6	258	13.93	3870.56	0.00
4n25d4h3	4	11,5,8,5	330	43.8	4925.73	0.00
5n25d4h3	4	9,9,9,2	343	9.88	4475.17	0.00
1n30d4h3	4	6,13,5,10	533	76.56	4262.32	0.00
2n30d4h3	4	5,14,6,9	562	22.2	3733.09	0.00
3n30d4h3	4	7,13,4,10	533	18.94	4649.11	0.00
4n30d4h3	4	11,11,1,11	608	18.26	358.22	0.00
5n30d4h3	4	11,13,5,5	560	21.9	4199.74	0.00
1n35d5h3	5	8,9,6,4,13	529	21.98	5351.66	0.00
2n35d5h3	5	11,7,7,4,11	531	28.02	5213.10	0.00
3n35d5h3	5	7,10,6,6,11	483	27.83	4790.49	0.00
4n35d5h3	5	11,10,5,3,11	581	31.33	5430.31	0.00
5n35d4h3	5	11,11,4,1,13	734	32.59	4545.17	0.00
1n40d5h3	5	11,4,3,17,10	1085	58.14	5810.59	0.00
2n40d5h3	5	10,7,8,10,10	598	57.39	5543.21	0.00
3n40d5h3	5	13,4,5,13,10	836	70.95	5568.42	0.00
4n40d5h3	5	13,4,7,13,8	798	49.24	5112.32	0.00
5n40d5h3	5	10,5,10,10,10	637	98.69	4695.32	0.00
1n45d6h3	6	15,4,3,15,2,12	654	136.24	6547.97	0.00
2n45d6h3	6	10,7,4,14,6,10	816	149.27	5896.87	0.00
3n45d6h3	6	10,4,7,10,10,10	683	69.24	6084.01	0.00
4n45d6h3	5	10,3,9,13,6,7	1059	47.41	6420.13	0.00
5n45d6h3	6	13,5,7,13,7,6	821	76.56	5002	0.00
1n50d6h3	6	7,16,16,5,2,10	1432	2047.59	7238.44	0.00
2n50d6h3	6	7,7,16,8,3,15	1302	197.92	6131.18	0.00
3n50d6h3	6	10,10,10,8,8,10	720	61.17	6788	0.00
4n50d6h3	6	8,7,13,4,11,13	999	70.88	6283.02	0.00
5n50d6h3	6	9,7,13,8,7,12	893	362.70	5914.12	0.00
Average			461.82	87.78		0.00

Table 3: Matheuristic performance for instances with $H = 3$

Instance	N. Clust.	$\|C_i\|$	N. Routes	Time (s)	Cost	MHIRP GAP%
1n5d2h6	2	4,3	19	0.17	2595.14	0.00
2n5d2h6	2	4,3	15	3.76	3296.32	0.00
3n5d2h6	2	4,3	18	0.8	5145.63	0.00
4n5d2h6	2	4,3	26	6.59	2963.12	0.00
5n5d2h6	2	4,3	26	86.09	3117.05	0.00
1n10d2h6	2	5,7	81	11.5	4363.89	0.00
2n10d2h6	2	5,7	86	369.07	5271.25	0.00
3n10d2h6	2	8,4	82	64.52	5290.70	0.00
4n10d2h6	2	4,8	115	665.89	5649.83	0.00
5n10d2h6	2	4,8	135	565.2	5052.26	0.00
1n15d2h6	2	7,10	239	3466.42	8942.68	0.00
2n15d2h6	2	10,7	233	1995.81	9072.41	0.00
3n15d2h3	2	8,9	260	1693.07	8775.99	0.00
4n15d2h6	2	12,5	272	5516.63	8690.27	0.00
5n15d2h6	2	9,8	236	473.50	890.,06	0.00
1n20d3h6	3	5,10,8	297	6614.13	9616.51	0.00
2n20d3h6	3	8,8,7	321	1596.92	8386.92	0.00
3n20d3h6	3	8,8,7	306	1405.85	10895.99	0.00
4n20d3h6	3	10,8,5	294	89.36	6826.68	0.00
5n20d3h6	3	13,9,1	495	1577.32	11343.97	0.00
1n25d4h6	4	15,7,2,5	617	9574	11276.2	0.00
2n25d4h6	4	9,7,5,8	397	3728.45	9172.41	0.00
3n25d4h6	4	9,8,6,6	251	92.81	9575.81	0.00
4n25d4h6	4	6,10,7,6	314	37.63	9215.81	0.00
5n25d4h6	4	8,10,6,5	353	538.80	8869.16	0.00
1n30d4h6	4	8,13,7,6	466	10800	12255.30	1.12
2n30d4h6	4	6,11,6,11	545	1575.14	12001.80	0.00
3n30d4h6	4	8,11,4,11	639	2542.20	10675.78	0.00
4n30d4h6	4	11,11,6,6	531	600.42	9875.90	0.00
5n30d4h6	4	11,11,6,6	536	2841.61	9325.37	0.00
1n35d5h6	5	9,10,10,4,7	501	61.94	11029.50	0.00
2n35d5h6	5	10,9,7,4,10	605	76.15	11195.46	0.00
3n35d5h6	5	8,10,6,6,10	651	1877.39	10747.05	0.00
4n35d5h6	5	10,8,10,6,6	435	71.3	10389.20	0.00
5n35d5h6	5	10,9,11,5,5	607	714.18	14750.82	0.00
1n40d5h6	5	11,8,11,10,5	763	1185.19	12152.40	0.00
2n40d5h6	5	10,7,8,10,10	812	4330.62	13797.56	0.00
3n40d5h6	5	10,5,10,10,10	865	4939.05	10840.55	0.00
4n40d5h6	5	14,8,10,6,7	790	124.50	12337.50	0.00
5n40d5h6	5	14,5,16,5,5	1149	2672.15	12673.98	0.00
1n45d6h6	6	7,13,8,11,6,6	761	576.78	10940.30	0.00
2n45d6h6	6	10,7,4,10,10,10	713	323.05	11323.10	0.00
3n45d6h6	6	7,8,8,13,7,8	680	640.15	11606	0.00
4n45d6h6	6	8,11,11,10,5,6	701	2236.47	11838.5	0.00
5n45d6h6	6	9,13,2,13,6,8	964	2836.15	12392.35	0.00
1n50d6h6	6	13,6,13,11,6,7	1049	5049.94	12892.30	0.00
2n50d6h6	6	10,10,10,8,8,10	954	1013.1	14529.04	0.00
3n50d6h6	6	10,10,10,8,8,10	918	2778.32	13245.80	0.00
4n50d6h6	6	16,11,16,2,4,7	1511	13186.52	13186.52	0.00
5n50d6h6	6	11,8,13,10,6,8	939	4175.72	12588.10	0.00
Average			492	1965.4		0.00

Table 4: Matheuristic performance for instances with $H = 6$

6.3 Performance comparison

In this section the comparison between the results of the matheuristic algorithm and the branch–and–cut is presented. Table 5 provides the results for all the instances with time horizon $H = 3$ and $H = 6$. The table is organized as follows. Column **Instance** provides the instance name. Columns **MH** and **TimeMH (s)** provide the cost of the matheuristic solution and the corresponding computational time, respectively. Columns **B&C** and **TimeB&C (s)** report the cost of the best branch–and–cut solution obtained within the time limit allowed and the corresponding computational time, respectively. Finally, column **Gap %** provides the gap between the two approaches, computed as

$$GAP = \frac{MH - B\&C}{B\&C} 100$$

Instance	MH	TimeMH (s)	B&C	TimeB&C (s)	Gap %	Instance	MH	TimeMH (s)	B&C	TimeB&C (s)	Gap %
1n5d2h3	1148.80	0.09	1148.80	89.71	0.00	1n5d2h6	2595.14	0.17	2595.14	8685.15	0.00
2n5d2h3	956.24	0.10	956.24	434.76	0.00	2n5d2h6	3296.32	3.76	3296.32	21700	0.00
3n5d2h3	1801.02	0.13	1801.02	360.51	0.00	3n5d2h6	5145.63	0.80	5145.63	21700	0.00
4n5d2h3	1425.39	0.12	1425.39	1537.42	0.00	4n5d2h6	2963.12	6.59	2963.12	21700	0.00
5n5d2h3	1808.40	0.08	1808.40	60.09	0.00	5n5d2h6	3117.05	86.09	3120.91	21700	-0.12
1n10d2h3	2112.53	1.76	2177.99	21700	-3.00	1n10d2h6	4363.89	11.5	5102.45	21700	-14,47
2n10d2h3	2425.97	2.21	2427.66	21700	-0.06	2n10d2h6	5271.25	369.07	5501.12	21700	-4.18
3n10d2h3	1651.54	0.83	1651.54	1854.78	0.00	3n10d2h6	5290.71	64.52	5617.17	21700	-5.81
4n10d2h3	2449.30	0.81	2449.30	1994.27	0.00	4n10d2h6	5649,83	665,89	5849,66	21700	-3,42
5n10d2h3	1982.90	0.89	1982.90	734	0.00	5n10d2h6	5052.26	565.20	5077.50	21700	-0.50
1n15d2h3	3891.56	3.76	4588.88	21700	-15.20	1n15d2h6	8942.69	3466.42	9904.03	21700	-9.71
2n15d2h3	2436.43	4.30	2436.43	20955	0.00	2n15d2h6	9072.41	1995.81	10202.70	21700	-11.08
3n15d2h3	3189.50	3.84	3075.69	6602.50	3.70	3n15d2h6	8775.07	1693.07	9276.72	21700	-5.41
4n15d2h3	2298.16	3.88	2481.13	21700	-7.37	4n15d2h6	8690.27	5516.63	8875.49	21700	-2.09
5n15d2h3	2329.42	1.93	2912.07	21700	-20.01	5n15d2h6	8905.06	473.50	8997.27	21700	-1.03
1n20d3h3	3095.91	11.66	3263.25	21700	-5.13	1n20d3h6	9616.51	6614.13	11433.40	21700	-15.89
2n20d3h3	4074.56	6.44	4395.27	21700	-7.29	2n20d3h6	8386.92	1596.92	9094.27	21700	-7.78
3n20d3h3	3361.74	5.79	3399.42	21700	-1.11	3n20d3h6	10895.99	1405.85	10807.75	21700	0.82
4n20d3h3	4151.87	7.88	4501.08	21700	-7.76	4n20d3h6	6826.68	89.36	12965.10	21700	-47.34
5n20d3h3	4235.63	27.81	4338.40	21700	-2.37	5n20d3h6	11343.97	1577.32	13545.80	21700	-16.25
1n25d4h3	3354.83	13.97	4203.45	21700	-20.19	1n25d4h6	11276.20	9574	11949.99	21700	-5.64
2n25d4h3	3654.25	11.07	4570.13	21700	-20.04	2n25d4h6	9172.41	3728.45	10369.70	21700	-11.54
3n25d4h3	3870.56	13.93	4127.82	21700	-6.23	3n25d4h6	9575.81	92.81	10511.3	21700	-8.90
4n25d4h3	4925.73	43.80	5722.99	21700	-13.93	4n25d4h6	9215.81	37.63	10457.10	21700	-11.87
5n25d4h3	4475.17	9.88	6214.81	21700	-27.99	5n25d4h6	8869.16	538.70	9556.95	21700	-7.20
1n30d4h3	4262.32	76.56	5206.49	21700	-18.13	1n30d4h6	12255.30	10800	1275.90	21700	-3.94
2n30d4h3	3733.09	22.20	4805.12	21700	-22.31	2n30d4h6	12001.80	1575.14	12848.94	21700	-6.59
3n30d4h3	4649.11	18.94	5344.53	21700	-13.01	3n30d4h6	10675.78	2542.20	11921.20	21700	-10.45
4n30d4h3	3580.22	18.26	4805.12	21700	-25.49	4n30d4h6	9875.90	600.42	11420.40	21700	-13.52
5n30d4h3	4199.74	21.90	4693.64	21700	-10.52	5n30d4h6	9325.37	2841.61	9960.20	21700	-6.37
1n35d5h3	5351.66	21.98	7013.75	21700	-23.70	1n35d5h6	11029.50	61.94	12924	21700	-14.66
2n35d5h3	5213.10	28.02	6138.55	21700	-15.08	2n35d5h6	11195.46	76.15	11561.10	21700	-3.16
3n35d5h3	4790.49	27.83	6228.25	21700	-23.08	3n35d5h6	10747.05	1877.39	11676.1	21700	-7.96
4n35d5h3	5430.31	31.33	6660.30	21700	-18.47	4n35d5h6	10389.2	71.3	11004.45	21700	-5.59
5n35d5h3	4545.17	32.59	5771.25	21700	-21.24	5n35d5h6	14750.82	714.18	15450.90	21700	-4.53
1n40d5h3	5810.59	58.14	7080.27	21700	-17.93	1n40d5h6	12152.40	1185.19	12208.50	21700	-0.46
2n40d5h3	5543.21	57.39	7128.72	21700	-22.24	2n40d5h6	13797.56	4330.62	1399.70	21700	-1.44
3n40d5h3	5568.42	70.95	6434.14	21700	-13.45	3n40d5h6	10840.55	4939.05	10887.50	21700	-0.43
4n40d5h3	5115.32	49.24	574,3	21700	-10.92	4n40d5h6	12337.50	124.50	12554.70	21700	-1.73
5n40d5h3	4695.32	98.69	5196.53	21700	-9.65	5n40d5h6	12673.9875	2672.15	13283.70	21700	-4.58
1n45d6h3	6547.97	136.24	7272.02	21700	-9.96	1n45d6h6	10940.30	576.78	10884.90	21700	0.51
2n45d6h3	5896.87	149.27	6634.67	21700	-11.12	2n45d6h6	11323.10	323.05	11596.40	21700	-2.53
3n45d6h3	6084.01	69.24	6721.43	21700	-9.48	3n45d6h6	11606	640.15	11639.20	21700	-0.29
4n45d6h3	6420.13	47.41	6292.32	21700	2.03	4n45d6h6	11838.5	2236.47	12166.64	21700	-2.29
5n45d6h3	5002	76.46	5650.86	21700	-11.48	5n45d6h6	12392.35	2836.15	13373.20	21700	-7.33
1n50d6h3	7238.44	2407.59	7881.37	21700	-8.16	1n50d6h6	12892.30	5049.94	14031.90	21700	-8.12
2n50d6h3	6131.18	197.92	7432.50	21700	-17.51	2n50d6h6	14529.04	1013.1	14635.36	21700	-0.73
3n50d6h3	6788	61.17	7611.76	21700	-10.82	3n50d6h6	13245.80	2778.32	13721.20	21700	-3.46
4n50d6h3	6238.02	70.88	7740.36	21700	-18.83	4n50d6h6	13122.22	4054.4	13202.47	21700	-0.60
5n50d6h3	5914.12	362.70	6509.11	21700	-9.14	5n50d6h6	12588.10	4175.72	13676.20	21700	-7.96
Average		87.78		18052.46	-10.47	Average		1965.40		21439.75	-5.98

Table 5: Performance comparison

The results shown in Tables 3 and 4 demonstrate that the matheuristic algorithm is significantly more effective than the branch–and–cut. It is clear that instances with a longer time horizon are more difficult to solve. Nevertheless, the results provided by the matheuristic are good for both data sets. The result demonstrates that the route generation phase is not affected by the time horizon dimension. For this reason, it is possible to solve the instances with $H = 6$ without increasing the number of route–variables in formulation (28)–(36). Instead, the branch–and–cut algorithm can solve only small and medium size instances, while the possibility to find optimal solutions decreases with the instance size. Futhermore, the branch–and–cut algorithm is effective only to solve single-vehicle and single-product IRP instances. The branch–and–cut algorithm is able to solve to optimality only 11 small instances. As a consequence, the time limit of 6 hours is reached in 89 instances.

In the set with $H = 3$, the average duality gap is equal to 31.24%, while in the set with $H = 6$ this gap is equal to 36.22%. There is no sensitive difference between the gaps of the two data sets. This is due to the good quality of the initial solution built with the procedure described in Section 4, and used as starting solution for the branch–and–cut algorithm. The availability of this good initial solution allows to reduce the number of nodes to explore in the branch–and–cut tree, mainly in the data set with $H = 6$. The drawback of the branch–and–cut is that in large instances all the computational time is spent adding cuts to the LP formulation at the root node. Table 5 shows the comparison of the matheuristic against the branch–and–cut. In 96% of all the cases, the matheuristic is able to find a solution in a smaller computational time than the one provided by the branch–and–cut algorithm. For the data set with $H = 3$, the matheuristic provides a feasible solution with a total cost lower than the one of the solution found by the branch–and–cut by 10.47%, on average. The matheuristic is able to find the best solution within a computational time that is 17964.68 seconds smaller on average than the ones of the branch–and–cut. For the data set with $H = 6$, the matheuristic finds a feasible solution

that is better by 5.98% on average than the best solution provided by the branch–and–cut, and in a computational time that is 19474.7 seconds smaller on average than the ones of the branch–and–cut.

7 Conclusion

In this study, the $MDIRP$ with a homogeneous fleet of vehicles is studied. While classical $IRPs$ have been studied extensively, the multi-depot case represents a variant not well investigated despite the possibility offered by this problem to model real cases in city logistics. A MILP formulation is presented. An effective matheuristic algorithm based on a new clustering phase to solve the $MDIRP$ is designed and implemented. The matheuristic was tested over two data sets of randomly generated instances with up to 50 customers, and compared with a branch–and–cut algorithm. The computational results show that the matheuristic is able to find better solution in a smaller computational time. Future research could be devoted to investigate possible improvements of the clustering phase in matheuristic, or to study the multi-product $MDIRP$ in order to better adapt to real cases.

References

Y. Adulyasak, J-F. Cordeau, and R. Jans. Formulations and branch-and-cut algorithms for multivehicle production and inventory routing problems. *INFORMS Journal of Computing*, 14:103–120, 2013.

D. Ahr. Contributions to multiple postmen problems, phd dissertation. *University of Heidelberg*, page 117, 2004.

H. Andersson, A. Hoff, M. Christiansen, G. Hasle, and A. Løkketangen. Industrial aspects

and literature survey: Combined inventory management and routing. *Computers & Operations Research*, 37(9):1515–1536, 2010.

J. Aráoz, E. Fernández, and O. Meza. Solving the prize-collecting rural postman problem. *European Journal of Operational Research*, 196(3):886–896, 2009.

C. Archetti, L. Bertazzi, G. Laporte, and M.G. Speranza. A branch-and-cut algorithm for a vendor-managed inventory-routing problem. *Transportation Science*, 41(3):382–391, 2007.

C. Archetti, N. Boland, and M.G. Speranza. A matheuristic for the multivehicle inventory routing problem. *INFORMS Journal on Computing*, 29(3):377–387, 2017.

F. Barahona and M. Grötschel. On the cycle polytope of a binary matroid. *Journal of Combinatorial Theory, Series B*, 40(1):40–62, 1986.

M. Barratt and A. Oliveira. Exploring the experiences of collaborative planning initiatives. *International Journal of Physical Distribution and Logistics Management*, 31(4):266–289, 2001.

T. Bektas, T. G. Crainic, and T. Van Woensel. From managing urban freight to smart city logistics networks. 2015.

W.J. Bell, L.M. Dalberto, M.L. Fisher, A.J. Greenfield, R. Jaikumar, P. Kedia, R.G. Mack, and P.J. Prutzman. Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. *Interfaces*, 13:4–23, 1983.

L. Bertazzi and M.G. Speranza. Continuous and discrete shipping stategies for the single link problem. *Transportation Science*, 36(3):314–325, 2002.

L. Bertazzi and M.G. Speranza. Matheuristics for inventory routing problems. *Hybrid*

algorithms for service, computing and manufacturing systems: routing and scheduling solutions. IGI Global, Hershey, pages 1–14, 2011.

L. Bertazzi and M.G. Speranza. Inventory routing with multiple customers. *EURO Journal on Transportation and Logistics*, 2:255–275, 2013.

L. Bertazzi, A. Bosco, F. Guerriero, and D. Laganà. A stochastic inventory routing problem with stock-out. *Transportation Research Part C: Emerging Technologies*, 27: 89–107, 2013.

L. Bertazzi, A. Bosco, and D. Laganà. Min–max exact and heuristic policies for a two-echelon supply chain with inventory and transportation procurement decisions. *Transportation Research Part E: Logistics and Transportation Review*, 93:57–70, 2016.

A.M. Campbell and M.W.P. Salvesbergh. A decomposition approach for the inventory routing problem. *Transportation Science*, 38(4):488–502, 2004.

M. Christiansen, K. Fagerholt, T. Flatberg, Ø. Haugen, O. Kloster, and E. H Lund. Maritime inventory routing with multiple products: A case study from the cement industry. *European Journal of Operational Research*, 208(1):86–94, 2011.

L.C. Coelho and G. Laporte. A branch-and-cut algorithm for the multi-product multi-vehicle inventory-routing problem. *International Journal of Production Research*, 51 (23-24):7156–7169, 2013.

L.C. Coelho, J.-F. Cordeau, and G. Laporte. Consistency in multi-vehicle inventory-routing. *Transportation Research Part C: Emerging Technologies*, 24:270–287, 2012.

L.C. Coelho, J.-F. Cordeau, and G. Laporte. Thirty years of inventory–routing. *Transportation Science*, 48:1–19, 2013.

L.C. Coelho, J.-F. Cordeau, and G. Laporte. Heuristics for dynamic and stochastic inventory-routing. *Computers & Operations Research*, 52:55–67, 2014.

J.-F. Cordeau, D. Laganà, R. Musmanno, and F. Vocaturo. A decomposition-based heuristic for the multiple-product inventory-routing problem. *Computers & Operations Research*, 55:153–166, 2015.

T.G. Crainic, N. Ricciardi, and G. Storchi. Models for evaluating and planning city logistics systems. *Transportation Science*, 43(4):432–454, 2009.

S. Dauzère-Pérès, A. Nordli, A. Olstad, K. Haugen, U. Koester, M. Per Olav, G. Teistklub, and A. Reistad. Omya Hustadmamor optimizes its supply chain for delivering calcium carbonate slurry to european paper manufacturers. *Interfaces*, 37(1):39–51, 2007.

G. Desaulniers, J. G. Rakke, and L. C. Coelho. A branch-price-and-cut algorithm for the inventory-routing problem. *Transportation Science*, 50(3):1060–1076, 2015.

M. Fischetti, J.J. Salazar González, and P. Toth. Solving the orienteering problem through branch-and-cut. *INFORMS Journal on Computing*, 10(2):133–148, 1998.

M. Gendreau, G. Laporte, and F. Semet. A branch-and-cut algorithm for the undirected selective traveling salesman problem. *Networks*, 32(4):263–273, 1998.

R. Grønhaug, M. Christiansen, G. Desaulniers, and J. Desrosiers. A branch-and-price method for a liquefied natural gas inventory routing problem. *Transportation Science*, 44(3):400–415, 2010.

Ç. Koç, T. Bektaş, O. Jabali, and G. Laporte. The impact of depot location, fleet composition and routing on emissions in city logistics. *Transportation Research Part B: Methodological*, 84:81–102, 2016.

M. Padberg and G. Rinaldi. Facet identification for the symmetric traveling salesman polytope. *Mathematical Programming*, 47(1-3):219–257, 1990.

D. Popović, M. Vidović, and G. Radivojević. Variable neighborhood search heuristic for the inventory routing problem in fuel delivery. *Expert Systems with Applications*, 39 (18):13390–13398, 2012.

S. Salhi, A. Imran, and N.A Wassan. The multi-depot vehicle routing problem with heterogeneous vehicle fleet: Formulation and a variable neighborhood search implementation. *Computers & Operations Research*, 52:315–325, 2014.

M.W.P. Savelsbergh and T. Van Woensel. City logistics: Challenges and opportunities. *Transportation Science*, 50(2):579–590, 2016.

Q. Shen, F. Chu, and H. Chen. A lagrangian relaxation approach for a multi-mode inventory routing problem with transshipment in crude oil transportation. *Computers & Chemical Engineering*, 35(10):2113–2123, 2011.

O. Solyalı and H. Süral. A single supplier–single retailer system with an order-up-to level inventory policy. *Operations Research Letters*, 36(5):543–546, 2008.

O. Solyalı, J.-F. Cordeau, and G. Laporte. Robust inventory routing under demand uncertainty. *Transportation Science*, 46(3):327–340, 2012.