Computers & Operations Research 182 (2025) 107084

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

# Multi-day routes in a multi-depot vehicle routing problem with intermediate replenishment facilities and time windows

Matteo Avolio [a], Massimo Di Francesco [b], Antonio Fuduli [a], Enrico Gorgone [b], Roberto Wolfler Calvo [b,c,*]

[a] Department of Mathematics and Computer Science, University of Calabria, Rende, Italy
[b] Department of Mathematics and Computer Science, University of Cagliari, Cagliari, Italy
[c] AOC, LIPN, University Paris13, Paris, France

ARTICLE INFO

*Keywords:*
Multi-depot vehicle routing problem
Intermediate facilities
Time windows
Multi-day routes
Arc-replenishment
Branch & price

ABSTRACT

We face a multi-depot vehicle routing problem with intermediate replenishment facilities and time windows (MDVRPITW), characterized by a heterogeneous fleet of vehicles. This paper proposes two formulations and an exact algorithm to solve the problem. The first formulation models the set of the intermediate replenishment facilities by an explicit set of nodes. The second formulation introduces replenishment arcs in place of these nodes, if a replenishment occurs between two successive nodes, and adopts standard arcs in case the load of a vehicle is not renewed. By suitable time windows associated with different days, both formulations easily allow the construction of multi-day routes, aimed at minimizing deadhead trips thanks to the possible storage of the freight in the vehicle depots between two consecutive workdays. To solve the problem, we propose a Branch & Price exact approach based on the second formulation. Some numerical experiments on real-type data show significant cost savings considering multi-day routes, compared to the single-day ones with deadhead movements from/to vehicle depots.

## 1. Introduction

This paper tackles a new multi-depot vehicle routing problem with intermediate replenishment facilities and time windows (MDVRPITW), which is inspired by the real case of a grocery transportation company operating in Sardinia (Italy). In this problem, the intermediate facilities are optional stops, as there is freedom in their choice, but they are mandatory to collect the load to be delivered (Schiffer et al., 2019). A fleet of vehicles is available to collect freight from the replenishment facilities and provide delivery services to a set of customers. Vehicles are heterogeneous and classified into different types, to account for their different depots, the different customers they can serve, and their different replenishment facilities, with possible overlaps among the various types. Each vehicle starts and ends its routes at its depot within a predefined time limit. The objective is to service all customers exactly once within their time window, minimizing the total routing costs.

At the beginning of the overall tour, the empty vehicles first stop at a replenishment facility to collect cargo for servicing customers in the surroundings. The empty vehicles in the middle of the workday are replenished at the same or a different facility. Before returning empty to the depot, the last stop of the day is usually a customer. However, some savings can be obtained by decreasing the deadhead trips with no load from/to the vehicle depots. Therefore, before returning to the depot, the last stop in a day may be also a replenishment facility, where vehicles can collect cargo to be delivered the next day. The opportunity of keeping freight in depots between two successive workdays increases the complexity of the routing problem, which cannot be separated into independent single-day problems. To our knowledge, this problem has not yet been faced in the vast literature on vehicle routing problems (Toth and Vigo, 2014).

The objective of this paper is to model and solve this new variant of the vehicle routing problem. We propose two different formulations of the problem on a directed graph $G = (V, E)$: a node-based model and an arc-replenishment model. In the first one, each replenishment facility is explicitly represented by a node that, for modelling simplicity, can be visited at most once without loss of generality: in fact, if needed, the same replenishment facility can be replicated by means of different nodes, whose maximum number can be easily determined in advance during a pre-processing phase. In the second formulation, vehicles can renew their load along the replenishment arcs (connecting two customers $i \in V$ and $j \in V$ or the vehicle depot $i \in V$ with a customer

---

* Corresponding author at: AOC, LIPN, University Paris13, Paris, France.
*E-mail addresses:* matteo.avolio@unical.it (M. Avolio), mdifrance@unica.it (M. Di Francesco), antonio.fuduli@unical.it (A. Fuduli), egorgone@unica.it (E. Gorgone), wolfler@lipn.univ-paris13.fr (R. Wolfler Calvo).

https://doi.org/10.1016/j.cor.2025.107084
Received 30 May 2024; Received in revised form 27 March 2025; Accepted 28 March 2025

$j$). Consequently, in such a case, each arc $(i,j) \in E$ plays a double role: as a standard arc if the load is not renewed between $i \in V$ and $j \in V$ and as a replenishment arc if a replenishment occurs between $i \in V$ and $j \in V$. As a result, the costs of the replenishment arcs take into account the position of the nearest intermediate replenishment facility (which we assume to have infinite capacity) with respect to the corresponding endpoints. As a consequence, the same replenishment facility can also be used between any pair of customers.

To the best of our knowledge, this problem setting has never been considered in the literature so far. Although days are not explicitly considered in the formulations we propose (as we will see in Section 3), both models allow to easily treat this case by simply setting the data in a suitable way by means of appropriate time windows associated with different days and of a suitable structure of the graph.

Since the arc-replenishment formulation is easier to solve, from the methodological point of view, this paper proposes a Branch & Price algorithm based on this formulation. The experiments show that the Branch & Price performs well. Its effectiveness allows to assess the savings that can be achieved in the construction of multi-day routes (with the possibility of keeping the cargo in the vehicle depots at the end of each workday), compared to independent single-day routes in which vehicles must return empty to their depots.

The paper is organized as follows. In Section 2, the related literature is reported. In Section 3 we describe the two different formulations of the problem: the node-based formulation and the arc-replenishment formulation. In Section 4 we describe the Branch & Price approach for solving the second formulation. In Section 5, the numerical experiments are reported on two types of instances: some literature instances, slightly modified to fit our problem, and a real type instance drawn from a database provided by the grocery distribution company that inspired this work. Finally, some conclusions are drawn in Section 6.

## 2. Related literature

This paper spans over several variants of the vehicle routing problem (VRP). One of the most distinguishing is the VRP with intermediate replenishment facilities, which allows drivers to continue making deliveries without necessarily returning to their depot. Therefore, in organizing the analysis of the literature, we chronologically refer to the papers in this area and point out which features have been introduced over time and the related methodological approaches, to address more complex problem settings.

The early papers in this date back to the eighties, when some heuristics were proposed for simple single-day problems with two depots and equal customer demands (Jordan, 1987; Jordan and Burns, 1984). The first exact approach was introduced by Bard et al. (1998), who proposed a Branch & Cut algorithm for solving a single-day VRP with capacity and route time constraints. Angelelli and Speranza (2002) investigated the use of intermediate facilities in periodic single depot VRP by a tabu search. Unlike in our problem setting, customers are supposed to be served according to a given frequency in the planning horizon using single-day routes. Crevier et al. (2007) proposed a heuristic for a single-day multi-depot vehicle routing problem with intermediate replenishment facilities, a homogeneous fleet of vehicles and customers without time windows. Tarantilis et al. (2008) proposed a framework of meta-heuristics for a single-day single-depot vehicle routing problem with a homogeneous fleet of vehicles. A complementary research was carried out in waste collection by Benjamin and Beasley (2010), who proposed metaheuristics for a single period VRP with flexibility in the choice of disposal facilities. A periodic VRP in this application area was faced in Hemmelmayr et al. (2013) by a heuristic approach.

The research on exact methods for the VRP with intermediate replenishment facilities is limited. For such a variant of the problem, a Branch & Price algorithm for the single-day case with a homogeneous fleet of vehicles was proposed in Muter et al. (2014), while Branch & Cut & Price algorithms were recently adopted in Marques et al. (2022) and Roboredo et al. (2023).

All together, our problem setting is a novelty in the VRP literature since: (a) the fleet of vehicles is heterogeneous and each customer cannot indifferently be serviced by any available vehicle; (b) the models we propose are easily adaptable, by suitable time windows, to generate multi-day routes useful to reduce eventual deadhead trips after servicing the last customer of the day.

From the methodological point of view, unlike the previous exact methods, in our Branch & Price approach the pricing subproblems refer to an arc-replenishment type formulation of the problem, introduced in Boland et al. (2000) and adopted also in Gianessi et al. (2016), i.e. to a formulation where each arc plays a double role: as a standard arc, in case a load of a vehicle is not renewed between two successive customers, and as a replenishment arc in case a replenishment occurs along the arc.

## 3. Problem modelling

In this section we introduce two different formulations aimed at modelling the VRP described at the beginning of Section 1 and characterized by time windows at each node. The relevant difference between these two formulations resides in modelling the replenishment facilities, that in the first model are represented by a subset of nodes and in the second model by special arcs, called replenishment arcs.

Since we admit the possibility to have multi-day trips, in both formulations the same vehicle depot of type $u$ is represented in the graph by three different nodes characterized by appropriate time windows and an appropriate structure in terms of arcs. In particular:

1. At the beginning of the overall time horizon, the depot is represented by a node $o_u$ without incoming arcs and characterized by outgoing arcs linked only with replenishment facilities. The vehicle starts (empty) its tour at this node and stops for the first time at a replenishment facility.
2. In the case of service between consecutive workdays, the depot is represented as a special customer requiring no demand and characterized by suitable time windows for each workday. At the end of each workday, the vehicle (either empty or with some cargo) stops at the corresponding node of this type.
3. At the end of the planning horizon, the depot is represented by a node $e_u$ without outgoing arcs and characterized by incoming arcs linked only with the actual customers. The vehicle terminates (empty) its tour at this node.

We recall that a relevant advantage in setting the time windows at each node is that eventual subtours in any solution are automatically forbidden.

### 3.1. The node-based formulation

In the node-based formulation, the problem is modelled by representing the replenishment facilities as explicit nodes of the graph.

Given a directed graph $G = (V, E)$ and $n$ types of vehicles, we set $N = \{1, \dots, n\}$. For each $u \in N$, we indicate by $K_u$ the set of vehicles of type $u$, by $C_u$ the set of customers that can be serviced by a vehicle of type $u$, by $D_u$ the set of the nodes representing the intermediate replenishment facilities for the vehicles of type $u$, and by $V_u$ and $E_u$ the sets of the nodes and the arcs relative to the vehicles of type $u$, respectively. In addition, we denote by $K = \bigcup_{u \in N} K_u$ the overall fleet of vehicles, by $C = \bigcup_{u \in N} C_u$ the set of all the customers and by $D = \bigcup_{u \in N} D_u$ the set of all the intermediate replenishment facilities. For each type $u$ we denote by $o_u$ and $e_u$, respectively, the nodes where the overall route of any vehicle of type $u$ begins and finishes, both of them representing practically the same depot of type $u$. Then we set $C^o_u = C_u \cup \{o_u\}$, $C^e_u = C_u \cup \{e_u\}$, $C^o = \bigcup_{u \in N} C^o_u$ and $C^e = \bigcup_{u \in N} C^e_u$.

As a consequence, we have

$$V_u = C_u \bigcup D_u \bigcup \{e_u\} \bigcup \{o_u\}$$

and

$$E_u = \{(i,j) \mid i \in V_u \setminus \{e_u\},\, j \in V_u \setminus \{o_u\},\, i \neq j\}. \tag{1}$$

The problem is characterized by the following data. For every $i \in C^e$, we indicate by $q_i$ the demand of the customer $i$ (assuming $q_{e_u} = 0$ for each $u \in N$) and, for each $u \in N$, we denote by $m_u$ and $Q_u$, respectively, the number and the capacity of the vehicles of type $u$. For every $i \in V$, $[a_i, b_i]$ is the time window of the node $i$ and $s_i$ is the service time, while, for every $(i,j) \in E$, $c^u_{i,j}$ is the route cost associated with the arc $(i,j)$ when it is traversed by a vehicle of type $u$. Finally, we denote by $d^u_{i,j}$ the time to traverse arc $(i,j)$ for vehicles of type $u$.

We consider three types of decision variables denoted by $x^k_{ij}$, $t^k_i$ and $f^k_i$ with the following meaning: $x^k_{ij}$ is equal to 1 if the arc $(i,j)$ is traversed by the vehicle $k$ and 0 otherwise, $t^k_i$ is the starting time of the service at the node $i$ visited by the vehicle $k$ and $f^k_i$ is the residual cargo of the vehicle $k$ immediately after visiting node $i$.

On the basis of the above notation and denoting by $u_k \in N$, for $k \in K$, the type of the vehicle $k$, the node formulation we propose is the following:

$$\min_{x,f,t} \sum_{k\in K} \sum_{(i,j)\in E_{u_k}} c^{u_k}_{ij} x^k_{ij} \tag{2}$$

$$\text{s.t.} \sum_{k\in K} \sum_{j\in V_{u_k}\setminus\{i\}} x^k_{ji} \geq 1, \quad i \in C \tag{3}$$

$$\sum_{k\in K} \sum_{j\in V_{u_k}\setminus\{i\}} x^k_{ji} \leq 1, \quad i \in D \tag{4}$$

$$\sum_{j\in D_{u_k}} x^k_{o_{u_k} j} \leq 1, \quad k \in K \tag{5}$$

$$\sum_{j\in V_{u_k}\setminus\{i\}} x^k_{ij} - \sum_{j\in V_{u_k}\setminus\{i\}} x^k_{ji} = 0, \quad k \in K,\, i \in C_{u_k} \cup D_{u_k} \tag{6}$$

$$\sum_{k\in K_u} \sum_{j\in D_u} x^k_{o_u j} \leq m_u, \quad u \in N \tag{7}$$

$$\sum_{k\in K} f^k_{o_{u_k}} + \sum_{k\in K} f^k_{e_{u_k}} = 0 \tag{8}$$

$$f^k_i \leq Q_{u_k} - q_i, \quad k \in K,\, i \in C_{u_k} \tag{9}$$

$$-M^{u_k}_1(1 - x^k_{ij}) \leq f^k_j - f^k_i + q_j, \quad k \in K, i \in C^o_{u_k},\, j \in C^e_{u_k} \tag{10}$$

$$M^{u_k}_1(1 - x^k_{ij}) \geq f^k_j - f^k_i + q_j, \quad k \in K, i \in C^o_{u_k},\, j \in C^e_{u_k} \tag{11}$$

$$t^k_{o_{u_k}} = a_{o_{u_k}}, \quad k \in K \tag{12}$$

$$a_i \leq t^k_i \leq b_i, \quad k \in K,\, i \in C^e_{u_k} \cup D_{u_k} \tag{13}$$

$$t^k_i + (s_i + d^{u_k}_{ij})x^k_{ij} - t^k_j \leq M^{ij}_2(1 - x^k_{ij}), \quad k \in K,\, (i,j) \in E_{u_k} \tag{14}$$

$$x^k_{ij} \in \{0,1\}, \quad k \in K,\, (i,j) \in E_{u_k} \tag{15}$$

$$f^k_i \geq 0, \quad k \in K,\, i \in V_{u_k}, \tag{16}$$

$$t^k_i \geq 0, \quad k \in K,\, i \in V_{u_k}, \tag{17}$$

where

$$M^u_1 = Q_u, \text{ for each } u \in N, \tag{18}$$

and

$$M^{ij}_2 = b_i - a_j, \text{ for each } (i,j) \in E, \tag{19}$$

play the role of big-$M$ parameters.

Some explanations on the model (2)–(17) are in order. The objective function (2) is aimed at minimizing the total routing cost of the fleet, while constraints (3) and (4) impose that each customer must be visited at least once and each replenishment facility at most once, respectively. If the delivery to a customer exceeds the capacity of some vehicle, we replicate the node representing the customer (in this case the same customer is actually visited more than once) and, whenever needed, the same strategy is adopted to replicate the intermediate replenishment facilities of type $u$ whose maximum number, determined in advance during a pre-processing phase, is equal to

$$\left\lceil \frac{\sum_{i\in C_u} q_i}{Q_u} \right\rceil.$$

In case of overlapping among the various types, the above formula is slightly modified taking into account the demands of all the involved customers (in the numerator) and the minimum capacity of the corresponding vehicles (in the denominator).

Note that, in correspondence to any optimal solution, due to the triangle inequality, constraint (3) are satisfied as equality.

Constraints (5)–(6) define the route structure (i.e. uniqueness and continuity) of the vehicles. In particular, if vehicle $k$ is used, its route must begin at the node $o_{u_k}$ and must terminate at the node $e_{u_k}$, since, by constraints (6), the customers and the replenishment facilities are necessarily intermediate nodes.

Constraints (7) impose that the number of used vehicles of type $u$ cannot exceed the upper bound $m_u$.

Constraints (8)–(11) manage the residual cargo of each vehicle during the trip, possibly using the intermediate replenishment facilities to renew the load when needed. In particular, constraint (8) forces each vehicle to depart and to return empty, while constraints (10)–(11) update the residual cargo of the vehicles when a delivery is performed. We observe that, if a delivery to a node $j$ is performed by vehicle $k$ immediately after the delivery to any node $i$, $x^k_{ij} = 1$ and, according to (10) and (11),

$$f^k_j = f^k_i - q_i. \tag{20}$$

Vice versa, in case a replenishment is performed immediately after servicing customer $i$, then $x^k_{ij} = 0$ making constraints (10) and (11) redundant.

We highlight that the refill of the flow is automatically recovered by reading the values assumed by variables $f^k_i$ and $f^k_j$ when a replenishment occurs after servicing customer $i$ and before servicing customer $j$.

Constraints (12)–(14) require the respect of the time windows $[a_i, b_i]$, which characterize each node $i$ of the graph. Apart from the trivial constraints (12) (concerning the initial nodes of the routes) and the standard time window constraints (13), constraints (14) impose that, in passing from node $i$ to node $j$, a vehicle $k$ must visit $j$ not before the time necessary to service $i$ and to traverse the arc $(i,j)$, possibly waiting in case of anticipate arrivals.

Finally we observe that the big-$M$ parameters $M^u_1$ and $M^{ij}_2$, defined by (18) and (19) and characterizing constraints (10), (11) and (14), are well posed due to the presence of constraints (9) and (13), respectively.

### 3.2. *The arc-replenishment formulation*

Unlike (2)–(17), where the replenishment facilities are represented by the subset $D$ of nodes, the arc-replenishment formulation models the replenishment by arcs with related costs and times (Boland et al., 2000).

Using the same notation adopted in the previous subsection, we have

$$V_u = C_u \bigcup \{e_u\} \bigcup \{o_u\},$$

with the consequent modification of $E_u$, defined by (1).

For $u \in N$ and $(i,j) \in E_u$, we indicate by $r^u_{ij}$ and $w^u_{ij}$ the replenishment cost and the corresponding traversing time, relative to a vehicle of type $u$ along arc $(i,j)$. We introduce also the additional decision variables $y^k_{ij} \in \{0,1\}$, with the following meaning: $y^k_{ij}$ is equal to 1 in case the vehicle $k$ performs a replenishment along arc $(i,j)$, and 0 otherwise. Then, the arc-replenishment model can be written as follows.

$$\min_{x,y,f,t} \sum_{k\in K} \sum_{(i,j)\in E_{u_k}} c_{ij}^{u_k} x_{ij}^k + \sum_{k\in K} \sum_{(i,j)\in E_{u_k}} r_{ij}^{u_k} y_{ij}^k \tag{21}$$

$$s.t. \sum_{k\in K} \sum_{j\in C_{u_k}^o\setminus\{i\}} (x_{ji}^k + y_{ji}^k) \geq 1, \quad i \in C \tag{22}$$

$$\sum_{j\in C_{u_k}} y_{o_{u_k}j}^k \leq 1, \quad k \in K \tag{23}$$

$$\sum_{j\in C_{u_k}^e\setminus\{i\}} x_{ij}^k - \sum_{j\in C_{u_k}^o\setminus\{i\}} y_{ji}^k + \sum_{j\in C_{u_k}\setminus\{i\}} (y_{ij}^k - x_{ji}^k) = 0, \quad k \in K,\ i \in C_{u_k} \tag{24}$$

$$\sum_{k\in K_u} \sum_{j\in C_u} y_{o_u j}^k \leq m_u, \quad u \in N \tag{25}$$

$$\sum_{k\in K} f_{o_{u_k}}^k + \sum_{k\in K} f_{e_{u_k}}^k = 0 \tag{26}$$

$$f_i^k \leq Q_{u_k} - q_i, \quad k \in K,\ i \in C_{u_k} \tag{27}$$

$$-M_1^{u_k}(1 - x_{ij}^k) \leq f_j^k - f_i^k + q_j, \quad k \in K, i \in C_{u_k}^o,\ j \in C_{u_k}^e \tag{28}$$

$$M_1^{u_k}(1 - x_{ij}^k) \geq f_j^k - f_i^k + q_j, \quad k \in K, i \in C_{u_k}^o,\ j \in C_{u_k}^e \tag{29}$$

$$t_{o_{u_k}}^k = a_{o_{u_k}}, \quad k \in K \tag{30}$$

$$a_i \leq t_i^k \leq b_i, \quad k \in K,\ i \in C_{u_k}^e \tag{31}$$

$$t_i^k + d_{ij}^{u_k} x_{ij}^k + w_{ij}^{u_k} y_{ij}^k + s_i(x_{ij}^k + y_{ij}^k) - t_j^k \leq$$

$$M_2^{ij}(1 - x_{ij}^k - y_{ij}^k), \quad k \in K,\ (i,j) \in E_{u_k} \tag{32}$$

$$x_{ij}^k \in \{0,1\}, \quad k \in K,\ (i,j) \in E_{u_k} \tag{33}$$

$$y_{ij}^k \in \{0,1\}, \quad k \in K,\ (i,j) \in E_{u_k} \tag{34}$$

$$f_i^k \geq 0, \quad k \in K,\ i \in V_{u_k} \tag{35}$$

$$t_i^k \geq 0, \quad k \in K,\ i \in V_{u_k} \tag{36}$$

Apart from the absence of the set $D$, which does not allow to manage the time windows on the replenishment facilities, model (21)–(36) is very similar to model (2)–(17) and no particular additional comment is needed. We only note that, in constraints (22) and (24), variables $x$ and $y$ are alternative for a customer $i$, which can be visited from $j$ with or without a replenishment: in particular, while constraints (22) force to service all the customers, constraints (24) are the degree constraints imposing that, once a customer $i$ is reached (possibly after a replenishment) along an incoming arc by a vehicle $k$, an outgoing arc from $i$ must be traversed (with or without replenishment).

## 4. A branch & price approach

The above models (2)–(17) and (21)–(36) are not directly exploitable from the practical point of view, in case we have to solve large instances. Then a common technique used for large scale VRPs is the Branch & Price approach (see for example (Feillet, 2010)) applied to a set covering type formulation of the problem.

Since, from the preliminary numerical results presented in Section 5.1, the arc-replenishment model has revealed to be much more performant than the classical one, for the sake of simplicity we will report the pricing subproblems only to the arc-replenishment model (21)–(36).

We start from the following set covering formulation of the problem:

$$\min_{z} \sum_{u\in N} \sum_{r\in R_u} c_r^u z_r^u \tag{37}$$

$$s.t. \sum_{u\in N} \sum_{r\in R_u} a_{ir}^u z_r^u \geq 1, \qquad i \in C \tag{38}$$

$$\sum_{r\in R_u} z_r^u \leq m_u, \qquad u \in N \tag{39}$$

$$z_r^u \in \{0,1\}, \qquad u \in N,\ r \in R_u, \tag{40}$$

where $R_u$ is the set of all the feasible routes of type $u$ (i.e. covered by a vehicle of type $u$), $c_r^u$ is the cost of the route $r$ of type $u$ and $a_{ir}^u$ is a binary quantity equal to 1 if the customer $i$ is visited by the route $r$ of type $u$, and 0 otherwise. For every $u \in N$ and $r \in R_u$, the decision variables $z_r^u$ have the following meaning: $z_r^u$ is equal to 1 if the route $r$ of type $u$ is selected, and 0 otherwise. Referring to model (21)–(36), we have

$$c_r^u = \sum_{(i,j)\in E_u} b_{ijr}^u c_{ij}^u + \sum_{(i,j)\in E_u} d_{ijr}^u r_{ij}^u \tag{41}$$

and

$$a_{ir}^u = \sum_{j|(i,j)E_u} b_{ijr}^u + \sum_{j|(i,j)E_u} d_{ijr}^u, \tag{42}$$

where $b_{ijr}^u$ [resp. $d_{ijr}^u$] is equal to 1 if the standard [resp. replenishment] arc $(i,j)$ belongs to the route $r$ of type $u$, and 0 otherwise.

A route of type $u$ is said to be feasible if it starts at $o_u$ and terminates at $e_u$, visiting each customer of type $u$ at most once and satisfying the demand and the time window constraints.

It is well known that solving problem (37)–(40) by a standard Branch & Bound approach is prohibitive from the computational point of view, since the continuous relaxations to be solved are characterized by a huge number of variables, due to the exponential number of feasible routes. Then a consolidated approach used to overcome this difficulty is to apply the Branch & Price algorithm, which is based on a column generation technique, consisting in firstly solving the continuous relaxation of problem (37)–(40), named *Master Problem* (MP), in correspondence to a subset of routes. In other words, we initially consider the following *Restricted Master Problem* (RMP):

$$\min_{z} \sum_{u\in N} \sum_{r\in \bar{R}_u} c_r^u z_r^u \tag{43}$$

$$s.t. \sum_{u\in N} \sum_{r\in \bar{R}_u} a_{ir}^u z_r^u \geq 1, \qquad i \in C \tag{44}$$

$$\sum_{r\in \bar{R}_u} z_r^u \leq m_u, \qquad u \in N \tag{45}$$

$$0 \leq z_r^u \leq 1, \qquad u \in N,\ r \in \bar{R}_u, \tag{46}$$

where $\bar{R}_u \subset R_u$, for $u \in N$.

An important role is played by the dual of RMP, whose construction could be simplified by avoiding the presence of the primal constraints $z_r^u \leq 1$. This is easily obtainable by imposing, in the original problem (37)–(40), $z_r^u$ nonnegative and integer instead of $z_r^u \in \{0,1\}$: this substitution can be adopted without loss of generality, since in any optimal solution $z_r^u$ cannot be greater than 1. Taking into account this consideration, the dual of RMP is the following:

$$\max_{\lambda,\mu} \sum_{i\in C} \lambda_i - \sum_{u\in N} m_u \mu_u \tag{47}$$

$$s.t. \sum_{i\in C} a_{ir}^u \lambda_i - \mu_u \leq c_r^u, \qquad u \in N,\ r \in \bar{R}_u \tag{48}$$

$$\lambda_i \geq 0, \qquad i \in C \tag{49}$$

$$\mu_u \geq 0, \qquad u \in N, \tag{50}$$

where $\lambda_i$, for $i \in C$, and $\mu_u$, for $u \in N$, are the dual variables associated with the primal constraints (44) and (45), respectively.

Given an optimal solution $(\lambda^*, \mu^*)$ of the dual program (47)–(50), the column generation approach is based on generating new routes to be added into the RMP and characterized by a negative reduced cost, i.e. routes $r$ of type $u$ such that

$$c_r^u - \sum_{i\in C} a_{ir}^u \lambda_i^* + \mu_u^* < 0. \tag{51}$$

In particular, from (41) and (42) condition (51) reduces to

$$\sum_{(i,j)\in E_u} b_{ijr}^u (c_{ij}^u - \lambda_i^*) + \sum_{(i,j)\in E_u} d_{ijr}^u (r_{ij}^u - \lambda_i^*) + \mu_u^* < 0.$$

Then, in order to generate new routes with negative reduced cost, it is sufficient to solve $n$ subproblems (one for each type $u \in N$), commonly named *pricing subproblems*, of the type:

$$obj^u_{price} = \min_{x,y,f,t} \sum_{(i,j)\in E_u} (c^u_{ij} - \lambda^*_i)x_{ij} + \sum_{(i,j)\in E_u} (r^u_{ij} - \lambda^*_i)y_{ij} + \mu^*_u \tag{52}$$

$$\sum_{j\in C_u} y_{o_u j} \leq 1 \tag{53}$$

$$\sum_{j\in C^e_u\setminus\{i\}} x_{ij} - \sum_{j\in C^o_u\setminus\{i\}} y_{ji} + \sum_{j\in C_u\setminus\{i\}} (y_{ij} - x_{ji}) = 0, \quad i \in C_u \tag{54}$$

$$f_{o_u} + f_{e_u} = 0 \tag{55}$$

$$f_i \leq Q_u - q_i, \quad i \in C_u \tag{56}$$

$$-M^u_1(1 - x_{ij}) \leq f_j - f_i + q_j, \quad i \in C^o_u,\ j \in C^e_u \tag{57}$$

$$M^u_1(1 - x_{ij}) \geq f_j - f_i + q_j, \quad i \in C^o_u,\ j \in C^e_u \tag{58}$$

$$t_{o_u} = a_{o_u} \tag{59}$$

$$a_i \leq t_i \leq b_i, \quad i \in C^e_u \tag{60}$$

$$t_i + d^u_{ij}x_{ij} + w^u_{ij}y_{ij} + s_i(x_{ij} + y_{ij}) - t_j \leq M^{ij}_2(1 - x_{ij} - y_{ij}), \quad (i,j) \in E_u \tag{61}$$

$$x_{ij} \in \{0,1\}, \quad (i,j) \in E_u \tag{62}$$

$$y_{ij} \in \{0,1\}, \quad (i,j) \in E_u \tag{63}$$

$$f_i \geq 0, \quad i \in V_u \tag{64}$$

$$t_i \geq 0, \quad i \in V_u, \tag{65}$$

where $x_{ij}$ [resp. $y_{ij}$] is equal to 1 if arc $(i,j)$ is selected as a standard [resp. replenishment] arc and 0 otherwise, $t_i$ is the starting time of the service at the node $i$ and $f_i$ is the residual cargo immediately after visiting node $i$.

The column generation algorithm, referred to the arc-replenishment model (21)–(36) and aimed at solving the continuous relaxation of problem (37)–(40), is summarized in Algorithm 1.

**Algorithm 1: Column generation**

```
                                              ▷Initialization
1 for $u \leftarrow 1, \dots, n$ do
2     Initialize $\bar{R}_u$
3 repeat
                                              ▷Solving (RMP)
4     Solve problem (43)-(46) and compute the dual variables
      $(\lambda^*, \mu^*)$
                              ▷Solving the pricing subproblems
5     for $u \leftarrow 1, \dots, n$ do
6         Let $\bar{r}$ be the route generated in correspondence to an
          optimal solution of the pricing subproblem (52)-(65)
7         if $obj^u_{price} < 0$ then
8             $\bar{R}_u \leftarrow \bar{R}_u \cup \{\bar{r}\}$
9 until $obj^u_{price} \geq 0$, for $u = 1, \dots, n$
```

It is easy to see that each pricing subproblem (52)–(65), to be solved at step 6 of Algorithm 1, is a variant of the well-known elementary shortest path problem with resource constraint (ESPPRC). Hence, to solve this subproblem, we adopted a modified version of the algorithm presented in Lozano et al. (2016). Since the same dominance rules apply unless a refuelling occurs at one of the last two current nodes on the path, in the event of a replenishment rollback cannot be performed due to the higher value of the current load.

Finally, about the initialization of the route sets $\bar{R}_1, \dots, \bar{R}_n$, we have used a simple strategy based on generating, for each customer, a high-cost route serving only that specific customer and covered by a vehicle of a fictitious type $n+1$, such that $m_{n+1} = +\inf$.

### 4.1. The branching strategy

Concerning the branching strategy, we have applied the following scheme, using the same notation adopted in (41) and (42). We first associate to any arc $(i,j) \in E$ a sort of priority value, expressed by

$$\min\{\sum_{u\in N}\sum_{r\in R_u} b^u_{ijr}z^u_r, 1 - \sum_{u\in N}\sum_{r\in R_u} b^u_{ijr}z^u_r\} \max_{u\in N} c^u_{ij}, \tag{66}$$

in case $(i,j)$ is considered a standard arc, and by

$$\min\{\sum_{u\in N}\sum_{r\in R_u} d^u_{ijr}z^u_r, 1 - \sum_{u\in N}\sum_{r\in R_u} d^u_{ijr}z^u_r\} \max_{u\in N} r^u_{ij}, \tag{67}$$

in case $(i,j)$ is considered a replenishment arc. Then, we perform the branching on an arc $(\bar{\imath},\bar{\jmath})$ such that the corresponding priority, expressed by formulae (66) or (67), is maximum. Note that, since each arc $(i,j)$ has different costs in function of the corresponding type $u$, the above formulae, roughly speaking, provide the maximum cost of arc $(i,j)$ with respect to all its types, weighted by a quantity which approximately depends on how many times the arc appears in the selected routes.

Once the arc $(\bar{\imath},\bar{\jmath})$ has been selected, we generate two subproblems. The first one is obtained by imposing that the customer $\bar{\jmath}$ must be serviced immediately after the customer $\bar{\imath}$, with or without replenishment depending on where the maximum priority between (66) and (67) is reached. Vice versa, the second subproblem is generated by imposing that the customer $\bar{\jmath}$ must not be serviced immediately after the customer $\bar{\imath}$ without replenishment, in case the maximum priority is obtained in correspondence to (66), or with replenishment in case the maximum priority is attained in correspondence to (67).

## 5. Numerical experiments

We have performed two types of numerical experiments, conducted respectively on two different classes of problem tests.

In the first type of experiments we have used some literature instances, slightly modified in order to fit our problem, modelled by the node-based (2)–(17) and the arc-replenishment (21)–(36) formulations. On the other hand, to test our problem also on real type data, we have conducted the second type of experiments on an additional instance, based on some data drawn from a database provided by the grocery transportation company who inspired this work.

In particular, while in using the literature instances we have considered the two cases of single-day and multi-day routes by suitably setting the time windows for each node, in the second type of experiments we have treated only the multi-day case, on which the real problem of the grocery transportation company is based.

We have implemented the resolution of both the models (2)–(17) and (21)–(36), solved directly by means of Gurobi (version 9.1.2), and the Branch & Price approach described in Section 4. For the latter, at each iteration, we have used Gurobi as well to solve the Restricted Master Problem (43)–(46), while, for the pricing subproblem (52)–(65), we have adopted a modified version of the Pulse Algorithm (Lozano et al., 2016), obtained by simply replicating the arcs along which a replenishment can occur.

The codes have been implemented in Julia (version 1.8.5) and run on a Windows 10 system, characterized by 16 GB of RAM and a 1.80 GHz Intel Core i7 processor. For each solved problem test, a time limit equal to 1800 s has been fixed.

About the implementation of the Branch & Price algorithm, in order to have an initial set of routes (lines 1–2 of Algorithm 1), we have generated fictitious routes having an infinite cost and servicing only one customer (a fictitious route for each customer), as mentioned in Section 4: the type assigned to such routes is an additional fictitious type $n+1$ characterized by an infinite number of vehicles. Moreover, at each node of the Branch & Price tree, we also solve the integer program (37)–(40), where $R_u$, $u \in N$, is substituted by the currently available subset $\bar{R}_u$ of routes to possibly improve the incumbent solution.

### 5.1. Literature benchmark instances

The first type of numerical experiments has been performed by using twenty MDVRPTW (multi-depot vehicle routing problem with time windows) instances, detailed in Cordeau et al. (2001) and listed in Table 1. For all such instances, distinguished into two different groups (a) and (b), the original number $K_{lit}$ of vehicles, the original number $C_{lit}$ of customers and the original number $D_{lit}$ of depots are reported. We recall that the main difference between groups (a) and (b) is the time windows of the customers, which in group (a) are narrow, whereas those ones in group (b) are larger. As a consequence, the latter is generally characterized by a smaller number of vehicles, since it is easier for a given vehicle to serve more customers.

As mentioned above, we have slightly modified such original instances in order to fit our problem, modelled by the node set and arc-replenishment formulations, represented by the mixed integer linear programs (2)–(17) and (21)–(36), respectively. Such modifications have mainly consisted in the introduction of a type for each depot, with the consequent specification also of the types for the vehicles and the customers, the latter possibly serviced by more than one type of vehicle. In particular, we have firstly set the number $n$ of types equal to $D_{lit}$ and, for each $u \in N = \{1, \dots, n\}$, we have fixed the number $K_u$ of vehicles of type $u$ equal to $K_{lit}$, resulting in a total number $K$ of vehicles equal to $D_{lit}K_{lit}$. Then, for each $u \in N$, among the original $D_{lit}$ depots, we have chosen one depot coinciding with $e_u$ and $o_u$ (which represent the starting–ending depot of vehicles of type $u$), utilizing the remaining $D_{lit} - 1$ depots for the set $D_u$ forming the intermediate replenishment facilities of type $u$. About the types to be assigned to the customers (i.e. the types of vehicles by which the customers can be serviced), they have been randomly selected in the set $N$, guaranteeing at least one type per customer.

Utilizing these settings, from the literature instances of Table 1 we have generated four groups of test problems in function of the number $C$ of customers: very small instances (obtained by selecting the first 10 customers and used only preliminarily, as specified below), small instances (obtained by considering the first 24 customers), medium instances (obtained by selecting the first 48 customers) and large instances (obtained by taking the first 72 customers). Among the large instances, we have obviously had to exclude problems 1a and 1b, since they contain only 48 customers.

In order to have an idea of the performance exhibited by the two formulations (2)–(17) and (21)–(36) (classical node and arc-replenishment models, respectively), we have preliminary tested both of them on the very small instances (10 customers) in generating single-day optimal routes, solving directly the two respective mixed integer programs by Gurobi. The obtained results, expressed in terms of CPU time, are reported in the last two columns (columns T) of Table 1, where we do not give any information about the reached objective function values, since all the instances have been solved to optimality. Observing the CPU times, it is evident that the arc-replenishment model performs much faster than the first one, which was expected, due to the absence, in the arc-replenishment model, of the node set $D$ representing explicitly the intermediate replenishment facilities. For this reason, from now on, all the results reported in the successive tables refer only to the arc-replenishment model (21)–(36), solved directly by Gurobi, and to the Branch & Price algorithm exploiting the column generation approach described in Algorithm 1.

In particular, we have tested the Branch & Price method on the small, medium, and large instances to first generate single-day optimal routes, in comparison with the direct resolution of the arc-replenishment model solved by Gurobi. The obtained results are detailed in Table 2, where we report the CPU times (columns T) and the gap values (columns G), the latter given by the difference between the best incumbent and the best lower bound. The expression “n.s.” means that no feasible solution has been found within the time limit of 1800 s.

**Table 1**
Original benchmark MDVRPTW instances (Cordeau et al., 2001) and preliminary comparison between the two proposed models on single-day very small benchmark instances (10 customers).

| Inst. | # veh. | # cust. | # depots | $C = 10$ | |
|---|---|---|---|---|---|
| | | | | Arc-repl. | Node |
| | $K_{lit}$ | $C_{lit}$ | $D_{lit}$ | T (s) | T |
| 1a | 2 | 48 | 4 | 0.04 | 1.07 |
| 2a | 3 | 96 | 4 | 0.04 | 1.40 |
| 3a | 4 | 144 | 4 | 0.12 | 10.62 |
| 4a | 5 | 192 | 4 | 0.03 | 4.73 |
| 5a | 6 | 240 | 4 | 0.09 | 3.97 |
| 6a | 7 | 288 | 4 | 0.04 | 6.55 |
| 7a | 2 | 72 | 6 | 0.01 | 0.52 |
| 8a | 3 | 144 | 6 | 0.01 | 0.89 |
| 9a | 4 | 216 | 6 | 0.01 | 2.80 |
| 10a | 5 | 288 | 6 | 0.02 | 1.31 |
| 1b | 1 | 48 | 4 | 0.12 | 5.34 |
| 2b | 2 | 96 | 4 | 0.03 | 1.80 |
| 3b | 3 | 144 | 4 | 0.15 | 67.37 |
| 4b | 4 | 192 | 4 | 0.75 | 244.91 |
| 5b | 5 | 240 | 4 | 0.09 | 682.98 |
| 6b | 6 | 288 | 4 | 0.12 | 14.93 |
| 7b | 1 | 72 | 6 | 0.02 | 3.84 |
| 8b | 2 | 144 | 6 | 0.18 | 238.26 |
| 9b | 3 | 216 | 6 | 0.07 | 11.07 |
| 10b | 4 | 288 | 6 | 0.24 | 378.99 |

Looking at the results, we observe that the Branch & Price algorithm is much faster concerning the direct resolution of model (21)–(36) by Gurobi. In general the instances of group (b) appear more difficult to be solved. Differently from Gurobi, the Branch & Price algorithm can always reach the optimality, except for the large instances of group (b) where, however, a gap less than 0.02 is always obtained within the time limit.

We conclude the subsection by describing the numerical experiments carried out in generating multi-day routes. Such experiments have been performed only on the large instances (72 customers), for which we report also in this case the comparison between the arc-replenishment model, solved by Gurobi, and the Branch & Price algorithm. In particular, considering a planning horizon of two and three days, we have operated in the following way. Once we have defined the types of vehicles in function of the original number $D_{lit}$ of depots (as previously explained at the beginning of the subsection), we have randomly assigned the types of vehicles by which each customer can be serviced. Successively, we have partitioned the 72 customers into three subsets of 24 customers, one subset per day: the first subset of customers must be serviced on the first day, the second subset on the second day and the third one on the third day. Then we have run Gurobi and the Branch & Price algorithm initially three times, in order to generate three single-day routes, one for each subset of customers. Successively, we have generated multi-day routes, firstly considering a planning horizon of two days and secondly a planning horizon of three days. In two-days and three-days routes, freight is allowed to be kept in the vehicle depots between consecutive days.

The corresponding numerical results are reported in Tables 3 and 4, relative to Gurobi and Branch & Price respectively. The columns Day 1, Day 2 and Day 3 concern single-day routes with deadhead trips from/to the depots of vehicle at beginning/end of workdays. The column Days 1&2 refers to two-days routes with deadhead trips from the depots of vehicle at the beginning of the first day and at the end of the second day. The column Days 1&2&3 concerns three-days routes with deadhead trips from the depots of vehicle at the beginning of the first day and at the end of the third day.

It is worth noting that the numerical results relative to the first group of 24 customers (second and third columns of Tables 3 and 4) are different from those ones reported in the columns with $C = 24$ of Table 2, due to the randomness by which the types of vehicles are assigned to each customer.

**Table 2**
Comparison between Gurobi and the Branch & Price algorithm on single-day small, medium and large benchmark instances (24, 48 and 72 customers, respectively).

| Inst. | $C = 24$ | | | | $C = 48$ | | | | $C = 72$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Solver | | B. & P. | | Solver | | B. & P. | | Solver | | B. & P. | |
| | T (s) | G | T | G | T | G | T | G | T | G | T | G |
| 1a | 0.26 | 0 | 1.06 | 0 | 18.11 | 0 | 1.99 | 0 | – | – | – | – |
| 2a | 0.78 | 0 | 0.12 | 0 | 18.11 | 0 | 1.99 | 0 | 1800 | 0.02 | 25.19 | 0 |
| 3a | 9.09 | 0 | 2.12 | 0 | 1626.84 | 0 | 10.48 | 0 | 1800 | 0.06 | 8.09 | 0 |
| 4a | 18.56 | 0 | 0.28 | 0 | 96.27 | 0 | 0.40 | 0 | 1800 | 0.04 | 28.19 | 0 |
| 5a | 0.93 | 0 | 0.07 | 0 | 1465.45 | 0 | 8.67 | 0 | 1800 | 0.05 | 16.98 | 0 |
| 6a | 3.88 | 0 | 0.10 | 0 | 245.52 | 0 | 10.68 | 0 | 1800 | 0.02 | 4.35 | 0 |
| 7a | 0.09 | 0 | 0.03 | 0 | 3.89 | 0 | 0.29 | 0 | 42.33 | 0 | 2.65 | 0 |
| 8a | 0.41 | 0 | 0.05 | 0 | 3.22 | 0 | 1.35 | 0 | 402.22 | 0 | 1.95 | 0 |
| 9a | 0.23 | 0 | 0.02 | 0 | 7.01 | 0 | 1.06 | 0 | 81.40 | 0 | 0.77 | 0 |
| 10a | 0.71 | 0 | 0.21 | 0 | 370.95 | 0 | 0.91 | 0 | 1800 | 0.05 | 122.07 | 0 |
| 1b | 1800 | 0.03 | 5.74 | 0 | 1800 | n.s. | 28.61 | 0 | – | – | – | – |
| 2b | 1800 | 0.01 | 1.53 | 0 | 1800 | n.s. | 398.11 | 0 | 1800 | n.s. | 1800 | 0.007 |
| 3b | 1800 | 0.17 | 0.60 | 0 | 1800 | n.s. | 8.25 | 0 | 1800 | n.s. | 1800 | 0.005 |
| 4b | 1555.34 | 0 | 1.45 | 0 | 1800 | 0.21 | 4.66 | 0 | 1800 | n.s. | 1436.73 | 0 |
| 5b | 1800 | 0.03 | 0.27 | 0 | 1800 | 0.18 | 753.10 | 0 | 1800 | n.s. | 564.53 | 0 |
| 6b | 1800 | 0.18 | 1.04 | 0 | 1800 | 0.29 | 928.88 | 0 | 1800 | n.s. | 1800 | 0.01 |
| 7b | 18.07 | 0 | 0.33 | 0 | 1800 | n.s. | 14.52 | 0 | 1800 | n.s. | 617.98 | 0 |
| 8b | 199.54 | 0 | 0.06 | 0 | 1800 | 0.23 | 12.56 | 0 | 1800 | n.s. | 1800 | 0.007 |
| 9b | 1800 | 0.05 | 0.28 | 0 | 1800 | 0.17 | 216.16 | 0 | 1800 | n.s. | 1800 | 0.02 |
| 10b | 570.75 | 0 | 0.29 | 0 | 1800 | 0.14 | 77.31 | 0 | 1800 | 0.29 | 1800 | 0.02 |

**Table 3**
Arc-replenishment model solved by Gurobi: Comparison in generating single-day and multi-day optimal routes on large benchmark instances (72 customers).

| Inst. | Day 1 | | Day 2 | | Day 3 | | Days 1&2 | | Days 1&2&3 | |
|---|---|---|---|---|---|---|---|---|---|---|
| $C = 72$ | T (s) | G | T | G | T | G | T | G | T | G |
| 1a | – | – | – | – | – | – | – | – | – | – |
| 2a | 0.89 | 0 | 0.28 | 0 | 2.30 | 0 | 9.23 | 0 | 775.81 | 0 |
| 3a | 1.55 | 0 | 0.85 | 0 | 0.56 | 0 | 5.46 | 0 | 15.09 | 0 |
| 4a | 0.66 | 0 | 1.41 | 0 | 0.55 | 0 | 32.65 | 0 | 79.41 | 0 |
| 5a | 0.53 | 0 | 4.18 | 0 | 3.19 | 0 | 12.48 | 0 | 443.06 | 0 |
| 6a | 1.61 | 0 | 14.21 | 0 | 1.59 | 0 | 147.58 | 0 | 1800 | 0.001 |
| 7a | 0.04 | 0 | 0.45 | 0 | 0.03 | 0 | 1.13 | 0 | 2.83 | 0 |
| 8a | 0.19 | 0 | 1.65 | 0 | 0.43 | 0 | 6.45 | 0 | 35.92 | 0 |
| 9a | 0.29 | 0 | 0.73 | 0 | 0.15 | 0 | 4.26 | 0 | 22.48 | 0 |
| 10a | 4.24 | 0 | 1.39 | 0 | 0.22 | 0 | 80.76 | 0 | 472 | 0 |
| 1b | – | – | – | – | – | – | – | – | – | – |
| 2b | 53.33 | 0 | 221.89 | 0 | 1800 | 0.05 | 1800 | 0.03 | 1800 | 0.08 |
| 3b | 1800 | 0.13 | 474.74 | 0 | 274.88 | 0 | 1800 | 0.15 | 1800 | 0.11 |
| 4b | 1800 | 0.08 | 1800 | 0.11 | 1800 | 0.04 | 1800 | 0.12 | 1800 | 0.13 |
| 5b | 1800 | 0.09 | 1800 | 0.10 | 1800 | 0.02 | 1800 | 0.14 | 1800 | 0.13 |
| 6b | 1800 | 0.15 | 1800 | 0.19 | 1800 | 0.02 | 1800 | 0.21 | 1800 | 0.18 |
| 7b | 40.79 | 0 | 715.10 | 0 | 177.81 | 0 | 1800 | 0.01 | 1800 | 0.03 |
| 8b | 1022.17 | 0 | 25.88 | 0 | 171.52 | 0 | 1800 | 0.04 | 1800 | 0.07 |
| 9b | 1527.33 | 0 | 24.31 | 0 | 81.54 | 0 | 1800 | 0.13 | 1800 | 0.12 |
| 10b | 260.35 | 0 | 1800 | 0.07 | 232.89 | 0 | 1800 | 0.13 | 1800 | 0.13 |
| **Average** | **561.89** | **0.03** | **482.62** | **0.03** | **452.65** | **0.01** | **916.66** | **0.05** | **1102.59** | **0.06** |

We observe that, apart from three instances in the multi-day case, the Branch & Price algorithm always reaches the optimality, while Gurobi is often not able to obtain an optimal solution on group (b) within the time limit. Moreover, if on one hand Branch & Price is once again confirmed as much faster than Gurobi, on the other hand the CPU time necessary to solve the multi-day case is in general much higher with respect to the single-day problems, which was expected due to a higher number of customers to be serviced in the multi-day routes.

In order to evaluate the practical impact of using multi-day routes, for each instance and along a planning horizon of two and three days, in Tables 5 and 6 we finally report the percentage objective function variation $\Delta$, computed as the relative difference between the sum of the costs of the single-day routes (second and fifth columns) and the cost of the corresponding multi-day routes (third and sixty columns). Looking at the results, it is evident that planning the deliveries along larger time horizons is in general more convenient at least in terms of cost, which will be also confirmed by the results presented in the next subsection on a real type instance.

### 5.2. A real type instance

In this subsection we describe the second type of numerical experiments, performed on a real type MDVRPITW instance.

The data characterizing such instance are drawn from a real-life database, which has been provided by a grocery distribution company operating in Sardinia.

Going into the detail, we have focused on servicing only the company's customers located in the centre-south of Sardinia, resulting in a number $C$ of customers equal to 69. The total number $K$ of available vehicles was 13, partitioned into seven different types, on the basis of the corresponding capacities and of the drivers' places (the depots). As done for the literature instances, the types of vehicle assigned to each customer have been randomly selected, guaranteeing to each customer at least one type. All the involved route costs have been computed as distances expressed in minutes.

Similarly to the literature instances, we have solved this problem by the Branch & Price algorithm, considering a planning horizon of two

**Table 4**

Branch & Price: Comparison in generating single-day and multi-day optimal routes on large benchmark instances (72 customers).

| Inst. | Day 1 | | Day 2 | | Day 3 | | Days 1&2 | | Days 1&2&3 | |
|---|---|---|---|---|---|---|---|---|---|---|
| $C = 72$ | T (s) | G | T | G | T | G | T | G | T | G |
| 1a | – | – | – | – | – | – | – | – | – | – |
| 2a | 1.71 | 0 | 0.07 | 0 | 0.32 | 0 | 8.83 | 0 | 582.64 | 0 |
| 3a | 0.11 | 0 | 0.03 | 0 | 0.16 | 0 | 1.35 | 0 | 29.57 | 0 |
| 4a | 0.07 | 0 | 0.05 | 0 | 0.06 | 0 | 1.11 | 0 | 38.43 | 0 |
| 5a | 0.10 | 0 | 0.23 | 0 | 0.13 | 0 | 3.08 | 0 | 339.91 | 0 |
| 6a | 0.42 | 0 | 0.11 | 0 | 0.05 | 0 | 12.82 | 0 | 300.79 | 0 |
| 7a | 0.02 | 0 | 0.79 | 0 | 0.02 | 0 | 1.34 | 0 | 7.28 | 0 |
| 8a | 0.12 | 0 | 2.55 | 0 | 0.03 | 0 | 9.46 | 0 | 80.43 | 0 |
| 9a | 0.05 | 0 | 0.09 | 0 | 0.02 | 0 | 1.35 | 0 | 12.20 | 0 |
| 10a | 0.48 | 0 | 0.17 | 0 | 0.03 | 0 | 13.09 | 0 | 114.64 | 0 |
| 1b | – | – | – | – | – | – | – | – | – | – |
| 2b | 1.14 | 0 | 1.90 | 0 | 0.55 | 0 | 25.92 | 0 | 1649.11 | 0 |
| 3b | 0.97 | 0 | 0.14 | 0 | 0.09 | 0 | 30.42 | 0 | 667.58 | 0 |
| 4b | 3.84 | 0 | 0.45 | 0 | 1.46 | 0 | 144.30 | 0 | 1800 | 0.008 |
| 5b | 3.56 | 0 | 0.49 | 0 | 0.59 | 0 | 1800 | 0.003 | No sol | |
| 6b | 0.29 | 0 | 0.50 | 0 | 0.25 | 0 | 120.02 | 0 | No sol | |
| 7b | 0.19 | 0 | 0.63 | 0 | 0.49 | 0 | 55.38 | 0 | 144.35 | 0 |
| 8b | 0.31 | 0 | 0.09 | 0 | 0.11 | 0 | 3.31 | 0 | 64.22 | 0 |
| 9b | 0.21 | 0 | 0.23 | 0 | 0.47 | 0 | 8.08 | 0 | 340.97 | 0 |
| 10b | 0.21 | 0 | 0.17 | 0 | 0.11 | 0 | 15.75 | 0 | 432.64 | 0 |
| **Average** | **0.77** | **0** | **0.49** | **0** | **0.28** | **0** | **125.32** | **0.0002** | **412.80** | **0.0005** |

**Table 5**

Arc-replenishment model solved by Gurobi: Comparison, in terms of objective function values, between single-day and multi-day optimal routes on large benchmark instances (72 customers).

| Inst. | Day 1+2 | Days 1 & 2 | | Day 1+2+3 | Days 1 &2 & 3 | |
|---|---|---|---|---|---|---|
| $C = 72$ | Obj | Obj | $\Delta$ | Obj | Obj | $\Delta$ |
| 1a | – | – | – | – | – | – |
| 2a | 3035.01 | 2921.52 | 0.04 | 4234.94 | 4025.70 | 0.05 |
| 3a | 3405.15 | 3106.62 | 0.09 | 5560.04 | 4836.52 | 0.13 |
| 4a | 3415.90 | 3238.03 | 0.05 | 4841.59 | 4536.71 | 0.06 |
| 5a | 2662.63 | 2530.83 | 0.05 | 3890.03 | 3703.37 | 0.05 |
| 6a | 2923.81 | 2803.98 | 0.04 | 4506.61 | 4200.02 | 0.07 |
| 7a | 3708.53 | 3607.34 | 0.03 | 5781.84 | 5551.60 | 0.04 |
| 8a | 3378.51 | 3246.91 | 0.04 | 5142.51 | 4868.39 | 0.05 |
| 9a | 3805.34 | 3724.69 | 0.02 | 5758.56 | 5585.38 | 0.03 |
| 10a | 3093.98 | 2956.99 | 0.04 | 4766.43 | 4513.53 | 0.05 |
| 1b | – | – | – | – | – | – |
| 2b | 2529.96 | 2451.62 | 0.03 | 3675.17 | 3556.80 | 0.03 |
| 3b | 2734.08 | 2646.91 | 0.03 | 4558.10 | 4336.24 | 0.05 |
| 4b | 2779.22 | 2676.94 | 0.04 | 3974.17 | 3841.14 | 0.03 |
| 5b | 2552.97 | 2493.06 | 0.02 | 3583.05 | 3464.96 | 0.03 |
| 6b | 2565.31 | 2479.08 | 0.03 | 3803.77 | 3753.25 | 0.01 |
| 7b | 3397.55 | 3286.71 | 0.03 | 5079.83 | 4793.19 | 0.06 |
| 8b | 2498.19 | 2464.61 | 0.01 | 4090.59 | 4035.37 | 0.01 |
| 9b | 3016.06 | 3014.02 | 0.001 | 4635.88 | 4605.08 | 0.01 |
| 10b | 2625.50 | 2561.43 | 0.02 | 3979.63 | 3877.63 | 0.03 |
| **Average** | | | **0.03** | | | **0.04** |

**Table 6**

Branch & Price: Comparison, in terms of objective function values, between single-day and multi-day optimal routes on large benchmark instances (72 customers).

| Inst. | Day 1+2 | Days 1 & 2 | | Day 1+2+3 | Days 1 &2 & 3 | |
|---|---|---|---|---|---|---|
| $C = 72$ | Obj | Obj | $\Delta$ | Obj | Obj | $\Delta$ |
| 1a | – | – | – | – | – | – |
| 2a | 3035.01 | 2921.52 | 0.04 | 4234.94 | 4025.70 | 0.05 |
| 3a | 3405.15 | 3106.62 | 0.09 | 5560.04 | 4836.52 | 0.13 |
| 4a | 3415.90 | 3238.03 | 0.05 | 4841.59 | 4536.71 | 0.06 |
| 5a | 2662.63 | 2530.83 | 0.05 | 3890.02 | 3703.37 | 0.05 |
| 6a | 2923.81 | 2803.98 | 0.04 | 4506.61 | 4200.02 | 0.07 |
| 7a | 3708.53 | 3607.34 | 0.03 | 5781.83 | 5551.60 | 0.04 |
| 8a | 3378.51 | 3246.91 | 0.04 | 5142.51 | 4868.39 | 0.05 |
| 9a | 3805.34 | 3724.69 | 0.02 | 5758.56 | 5585.38 | 0.03 |
| 10a | 3093.98 | 2956.99 | 0.04 | 4766.43 | 4513.53 | 0.05 |
| 1b | – | – | – | – | – | – |
| 2b | 2529.96 | 2451.62 | 0.03 | 3672.55 | 3537.11 | 0.04 |
| 3b | 2734.08 | 2570.31 | 0.06 | 4558.10 | 4334.83 | 0.05 |
| 4b | 2765.73 | 2669.24 | 0.03 | 3956.01 | 3772.07 | 0.05 |
| 5b | 2509.53 | 2417.33 | 0.04 | 3539.61 | No sol | – |
| 6b | 2538.94 | 2430.74 | 0.04 | 3777.40 | No sol | – |
| 7b | 3397.55 | 3286.71 | 0.03 | 5079.83 | 4776.78 | 0.06 |
| 8b | 2498.19 | 2464.61 | 0.01 | 4090.59 | 4035.24 | 0.01 |
| 9b | 3016.06 | 2970.16 | 0.02 | 4635.88 | 4487.77 | 0.03 |
| 10b | 2625.50 | 2552.89 | 0.03 | 3979.63 | 3860.88 | 0.03 |
| **Average** | | | **0.04** | | | **0.05** |

and three days and splitting the set of customers into three groups of 23 customers per day. The results, in terms of objective function value expressed in minutes, are reported in Table 7, where the comparison between the single-day routes and the multi-day routes is performed by computing the quantity $\Delta_{obj}$, obtained as the difference between the sum of the single-day route costs and the corresponding multi-day route cost.

We observe that in both the cases (two and three days) a remarkable saving in terms of costs is obtained when the planning of the routes is globally performed along the overall time horizon.

## 6. Conclusions

Inspired by a grocery distribution company operating in Sardinia (Italy), we have tackled a multi-depot vehicle routing problem with intermediate facilities and time windows. The main novelty characterizing this problem is the introduction of multi-day routes aimed at minimizing eventual deadhead trips after servicing the last customer of the day. This is a relevant development, in that the literature on vehicle routing problems has not investigated a problem with this combination of characteristics and, thus, has not offered proper tools to evaluate the impact multi-day routes on distribution costs. For this problem, we have proposed two different formulations: a node-based formulation and an arc-replenishment formulation. Since the latter has revealed to be easier to solve by Gurobi, in order to face large scale instances, we have designed a Branch & Price algorithm based on the arc-replenishment formulation. Numerical experiments have shown that the Branch & Price algorithm can effectively solve benchmark instances and a real type instance. In turn, this allows to perform experiments in order to quantify the savings that can be obtained by allowing the storage of freight in the depot of vehicles between consecutive workdays, as opposed to the single-day routes with deadhead movements from/to vehicle depots. As a future development, we aim to solve larger problems either in terms of number of days, customers,

**Table 7**
Branch & Price: Comparison, in terms of minutes, between single-day and multi-day optimal routes on a real type instance.

| Day 1 | Day 2 | Day 3 | Day 1+2 | Days 1&2 | | Day 1+2+3 | Days 1&2&3 | |
|---|---|---|---|---|---|---|---|---|
| Obj | Obj | Obj | Obj | Obj | $\Delta_{obj}$ | Obj | Obj | $\Delta_{obj}$ |
| 1189.66 | 1062.16 | 1336.49 | 2251.82 | 2116.61 | 135.21 | 3588.31 | 3386.30 | 202.01 |

vehicles and number of customers served by each vehicle. Owing to the problem size, we expect to switch to metaheuristics.

## CRediT authorship contribution statement

**Matteo Avolio:** Software, Methodology, Data curation, Writing – review & editing. **Massimo Di Francesco:** Funding acquisition, Conceptualization, Writing – review & editing. **Antonio Fuduli:** Writing – original draft, Methodology, Conceptualization, Writing – review & editing. **Enrico Gorgone:** Writing – review & editing, Conceptualization. **Roberto Wolfler Calvo:** Writing – review & editing, Supervision, Methodology, Conceptualization.

## Acknowledgements

This study was carried out within the MOST – Sustainable Mobility National Research Center and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.4 – D.D. 1033 17/06/2022, CN00000023). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

## Data availability

Data will be made available on request.

## References

Angelelli, E., Speranza, M.G., 2002. The periodic vehicle routing problem with intermediate facilities. European J. Oper. Res. 137 (2), 233–247.

Bard, J., Huang, L., Dror, M., Jaillet, P., 1998. A branch and cut algorithm for the VRP with satellite facilities. IIE Trans. 30 (9), 821–834.

Benjamin, A., Beasley, J., 2010. Metaheuristics for the waste collection vehicle routing problem with time windows, driver rest period and multiple disposal facilities. Comput. Oper. Res. 37 (12), 2270–2280.

Boland, N.L., Clarke, L.W., Nemhauser, G.L., 2000. The asymmetric traveling salesman problem with replenishment arcs. European J. Oper. Res. 123 (2), 408–427.

Cordeau, J.-F., Laporte, G., Mercier, A., 2001. A unified tabu search heuristic for vehicle routing problems with time windows. J. Oper. Res. Soc. 52 (8), 928–936.

Crevier, B., Cordeau, J.-F., Laporte, G., 2007. The multi-depot vehicle routing problem with inter-depot routes. European J. Oper. Res. 176 (2), 756–773.

Feillet, D., 2010. A tutorial on column generation and branch-and-price for vehicle routing problems. 4OR 8 (4), 407–424.

Gianessi, P., Ceselli, A., Létocart, L., Wolfler Calvo, R., 2016. A branch & price & cut algorithm for the vehicle routing problem with intermediate replenishment facilities. Electron. Notes Discrete Math. 55, 93–96.

Hemmelmayr, V., Doerner, K.F., Hartl, R.F., Rath, S., 2013. A heuristic solution method for node routing based solid waste collection problems. J. Heuristics 19 (2), 129–156.

Jordan, W.C., 1987. Truck backhauling on networks with many terminals. Transp. Res. B 21 (3), 183–193.

Jordan, W.C., Burns, L.D., 1984. Truck backhauling on two terminal networks. Transp. Res. B 18 (6), 487–503.

Lozano, L., Duque, D., Medaglia, A.L., 2016. An exact algorithm for the elementary shortest path problem with resource constraints. Transp. Sci. 50 (1), 348–357.

Marques, G., Sadykov, R., Dupas, R., Deschamps, J.-C., 2022. A branch-cut-and-price approach for the single-trip and multi-trip two-echelon vehicle routing problem with time windows. Transp. Sci. 56 (6), 1598–1617.

Muter, I., Cordeau, J.-F., Laporte, G., 2014. A branch-and-price algorithm for the multidepot vehicle routing problem with interdepot routes. Transp. Sci. 48 (3), 425–441.

Roboredo, M., Sadykov, R., Uchoa, E., 2023. Solving vehicle routing problems with intermediate stops using VRPSolver models. Networks 81 (3), 399–416.

Schiffer, M., Schneider, M., Walther, G., Laporte, G., 2019. Vehicle routing and location routing with intermediate stops: A review. Transp. Sci. 53 (2), 319–343.

Tarantilis, C., Zachariadis, E., Kiranoudis, C., 2008. A hybrid guided local search for the vehicle-routing problem with intermediate replenishment facilities. INFORMS J. Comput. 20 (1), 154–168.

Toth, P., Vigo, D., 2014. Vehicle Routing: Problems, Methods, and Applications. SIAM.