CIRRELT

CIRRELT-2025-03

Last-mile Delivery with Multiple Deliverymen: Formulation and Exact Solution Methods for a Two-echelon Vehicle Routing Problem

Fernando Senna
Leandro C. Coelho
Reinaldo Morabito
Pedro Munari

January 2025

Document de travail également publié par la Faculté des sciences de l'administration de l'Université Laval, sous le numéro FSA-2025-002

Bureau de Montréal
Université de Montréal
C.P. 6128, succ. Centre-Ville
Montréal (Québec) H3C 3J7
Tél : 1-514-343-7575
Télécopie : 1-514-343-7121

Bureau de Québec
Université Laval,
2325, rue de la Terrasse
Pavillon Palasis-Prince, local 2415
Québec (Québec) G1V 0A6
Tél : 1-418-656-2073
Télécopie : 1-418-656-2624

Last-mile Delivery with Multiple Deliverymen: Formulation and Exact Solution Methods for a Two-echelon Vehicle Routing Problem

Fernando Senna[1,2], Leandro C. Coelho[2,*], Reinaldo Morabito[1], Pedro Munari[1]

1. Industrial Engineering Department, Federal University of São Carlos, Brazil
2. Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Canada research chair in integrated logistics, Université Laval, Canada

Abstract. There is an increasing demand for cost- and time-efficient last-mile delivery due to the expansion of home-delivery systems. To respond to this need, many companies and academics have focused on inventive delivery schemes to reduce costs and improve the service level offered to customers. These efforts include integrating vehicles with walking carriers, which additionally dodges traffic and avoids an increase in the emission of greenhouse gases and other pollutants. The vehicle routing problem with time windows and multiple deliverymen models an example of such delivery systems. In this problem, each vehicle may travel with more than one deliveryman to serve more customers with each stop of the vehicle and reduce the overall time that the vehicle stays parked. As originally defined, this problem considers that the customers served from each parking location and the routes traveled by the deliverymen are predefined. We propose a variant of this problem in which both of these decisions can be optimized. The novel problem is formally defined and formulated. Theoretical properties and useful lower bounds are introduced and used to propose several valid inequalities. The problem is also decomposed in a Benders scheme and solved exactly by a branch-and-Benders-cut algorithm. Extensive computational experiments show the suitability of the proposed methodology to solve the problem. Furthermore, managerial insights indicate that the inclusion of the customer clustering and deliveryman routes in the optimization leads to an average cost reduction of over 10%, with this value being much higher for some instances.

Keywords: routing, last-mile delivery, multiple deliverymen, Benders decomposition, parking

Acknowledgements. This work was supported by the São Paulo Research Foundation (FAPESP) [grant numbers 2013/07375-0, 2021/14441-5, 2022/09679-5, 2022/05803-3], the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) [Finance Code 001], the National Council for Scientific and Technological Development (CNPq) [grant number 405702/2021-3, 304618/2023-3, 314079/2023-8], and the Natural Sciences and Engineering Research Council of Canada [grant number 2019-00094]. This support is greatly acknowledged. We thank the Digital Research Alliance of Canada for providing high-performance parallel computing facilities.

---

* Corresponding author: leandro.coelho@fsa.ulaval.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2025

1. Introduction

Last-mile delivery is a growing concern in logistics operations due to the increasing demand for efficient deliveries in cities caused by the urban population growth and the expansion of e-commerce (Bayliss et al., 2023). Compared to traditional routing problems, last-mile delivery encompasses additional challenges such as finding places to park and poor traffic conditions (Martinez-Sykora et al., 2020). Also, some cities have restrictions on vehicle sizes and circulation. Poorly designed delivery systems negatively affect the traffic and can lead to higher emission of greenhouse gases (GHGs) and other pollutants (Bektaş and Laporte, 2011). These questions show the importance of evaluating and designing more effective last-mile delivery systems.

A common approach in these systems is to use two-echelon schemes (Cuda et al., 2015; Sluijk et al., 2023), in which larger vehicles take the goods from the depot to transshipment facilities (satellites) and smaller vehicles take them from these facilities to the final customers. The main examples are the two-echelon vehicle routing problem (2E-VRP) and the two-echelon location routing problem (2E-LRP).

Another possibility is to use two-echelon schemes without having these transshipment satellites by using the first-echelon vehicles as mobile facilities and having smaller vehicles taking goods from the vehicles to the customers. Some common applications include having the customers served by drones (Amine Masmoudi et al., 2022), robots (Alfandari et al., 2022), or carriers on bicycles or walking (Cabrera et al., 2022; Bayliss et al., 2023; Senna et al., 2024a). These smaller vehicles take the goods directly from the vehicles to the customers, with no need for transshipment facilities while increasing the efficiency of the deliveries. This is highly beneficial since it does not incur in additional costs of facility location (2E-LRP) and transshipment (2E-VRP), and does not require infrastructure investments in satellites. Additionally, the use of greener options in the second echelon (drones, robots, and people walking or cycling) leads to reducing the emission of GHGs and pollutants and does not impact the traffic.

The vehicle routing problem with time windows and multiple deliverymen (VRPTWMD) models an interesting application that emerges in this context. Specifically, this problem addresses the case in which vehicles take goods from the depot to the customers and, once parked, the deliverymen traveling with this vehicle serve the customers. Each vehicle may carry more than one deliveryman and, in such case, they serve the customers in parallel, reducing the time that the vehicle stays parked throughout the route. Since vehicle costs are often higher than deliveryman costs, this creates an opportunity for cost reduction (Pureza et al., 2012).

The common approach in the VRPTWMD is to assume that the deliveryman routes and the definition of which customers are to be served from each parking location (clustering) can be preprocessed (Pureza et al., 2012; Álvarez and Munari, 2017; Munari and Morabito, 2018; De La Vega et al., 2020). The customer clustering has been addressed by Senarclens de Grancy and Reimann (2015) and Senarclens de Grancy (2015), and the deliveryman routes by Senna et al. (2024a), all of them showing the benefits of including these decisions on the problem. However, to the best of our knowledge, no other work has evaluated the impact of including both the deliveryman routes and the customer clustering in the optimization problem. In this paper, we extend the VRPTWMD by considering both of these decisions. The contributions of this paper are sixfold:

- The introduction of a variant of the VRPTWMD that involves the decision of which customers are to be served by a vehicle parked at each parking location and the deliveryman routes;

- The proposition of a compact mixed-integer programming formulation to represent this problem;
- The discussion of theoretical properties of the problem and the proposition of valid inequalities;
- A branch-and-Benders-cut algorithm to optimally solve the problem based on Benders decomposition;
- Extensive computational experiments evaluating the performance of the proposed solution approaches (compact formulation, valid inequalities, and branch-and-Benders-cut);
- Managerial insights that highlight the importance of including these decisions in the problem.

The remainder of this paper is structured as follows. In Section 2, a brief literature review of the VRPTWMD is presented. Section 3 defines the problem. In Section 4, the problem is formulated, some properties are discussed, and valid inequalities are proposed. In Section 5, a Benders decomposition is proposed along with a branch-and-Benders-cut algorithm. Section 6 presents the computational experiments and provides some interesting managerial insights. Section 7 discusses concluding remarks.

2. Literature review

The VRPTWMD was proposed by Pureza et al. (2012) to reflect the deliveries of a beverage company to a densely populated area. In large cities, it is common that vehicles spend long times in their deliveries traveling slowly (due to traffic) in search of a place to park. It is also common that there are many customers close to each other, creating an opportunity of serving such customers with a single stop of the vehicle. Although reducing the issues of traffic and lack of parking location availability, this approach has the downside of having the vehicles parked during long periods while a single deliveryman (the driver) serves many customers. To speed up the delivery process, this company came up with the idea of including more than one deliveryman in each vehicle, reducing the time that the vehicles stay parked and increasing the delivery efficiency. Since the costs associated with vehicles are usually higher than those of deliverymen, this creates an opportunity for cost reduction. Moreover, since deliverymen emit less GHGs and other pollutants, this business model has the beneficial side effect of reducing emissions.

Based on the operations of the beverage company studied, Pureza et al. (2012) defined the problem with two simplifying hypotheses: (i) the definition of the customers to be served by each vehicle stop (clusters) is predefined, and (ii) the deliverymen routes inside each cluster can be defined in a preprocessing phase.

Since then, most works that studied the VRPTWMD followed these ideas. Álvarez and Munari (2016) compared the performance of different metaheuristics to solve the problem. Souza Neto and Pureza (2016) extended the problem to include the possibility of multiple trips for the vehicles. Munari and Morabito (2018) proposed the first exact algorithm for the VRPTWMD, which is a branch-and-price method, and thus, it relies on the column generation technique. Álvarez and Munari (2017) combined this method with two metaheuristics to create a hybrid exact algorithm. De La Vega, Munari and Morabito looked at the problem with uncertainties by means of robust optimization heuristically (De La Vega et al., 2019) and exactly (De La Vega et al., 2020).

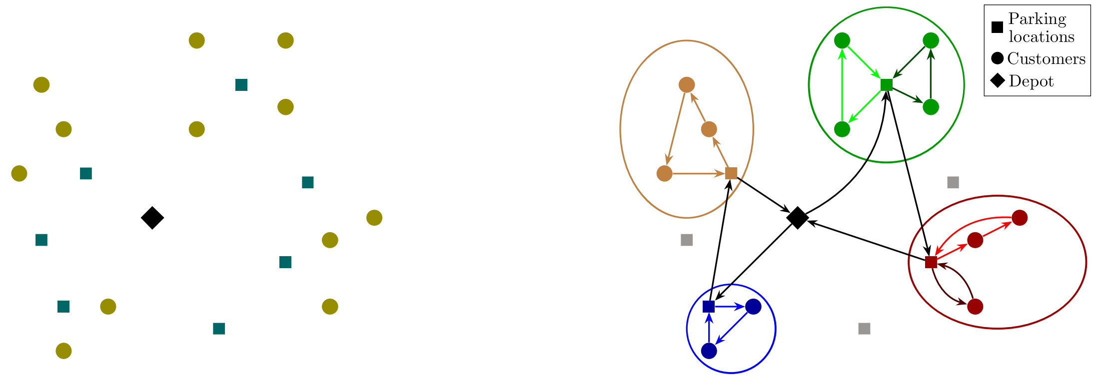
These hypotheses make sense in beverage delivery schemes, since the goods to be transported are usually large and heavy. Thus, a walking deliveryman cannot travel far from the vehicle while transporting these commodities, and the clusters can be easily defined based on customers that have compatible time windows and are very close to each other. Furthermore, the walking deliveryman would not be capable of serving many customers without coming back to the vehicle to collect more goods before heading to the next customer. This way, the deliveryman routes become trivial as back and forth trips from the vehicle to the customers.

Nonetheless, in different applications that include smaller demands or larger deliveryman capacities, the definition of customer clusters and deliveryman routes are not so straightforward and their inclusion in the optimization problem becomes beneficial. Senarclens de Grancy and Reimann (2015) were the first to realize this and propose the inclusion of the clustering in the problem. In fact, they proposed two novel heuristics to define the clusters. Senarclens de Grancy (2015) went further to combine the clustering with the routing. Both of these works looked at the VRPTWMD by removing the hypothesis (i) of predefined clustering while maintaining the hypothesis (ii) that the deliveryman routes are predefined. Senna et al. (2024a) looked at the problem from a different perspective, by evaluating the deliveryman routes and, hence, removing hypothesis (ii) that they should be preprocessed. However, they still considered that the clusters would be defined in a preprocessing phase as stated by hypothesis (i). These three studies proved the relevance of extending the VRPTWMD in these ways and the benefits it creates. Nevertheless, there is no work that addressed removing both of these simplifying hypotheses to include the customer clustering and the deliveryman routes in the optimization problem.

In this paper, we aim at bridging this gap by defining the vehicle routing problem with time windows, multiple deliverymen, customer clustering, and two-level routing (VRPTWMD-C2R). The VRPTWMD-C2R extends the VRPTWMD by including both the customer clustering and the deliveryman routes in the optimization process. As discussed above, this is especially interesting when having deliverymen with large capacities compared to customer demands. Moreover, when considering time windows, this becomes even more important. In previous approaches, clusters and/or deliveryman routes were preprocessed, requiring the order of visits within a cluster to be predefined. Consequently, the time windows of the parking locations were also preprocessed to ensure that these deliveryman routes aligned with the time windows of the customers and the predefined deliveryman routes. This preprocessing imposed additional constraints on the problem, potentially leading to suboptimal solutions. Notably, the VRPTWMD-C2R addresses this limitation by eliminating the need for such preprocessing.

3. Problem definition

The VRPTWMD-C2R is defined over a directed graph $G = (N, A)$, with N representing the set of nodes and A the set of arcs. Let N^1 be the set of the n potential parking locations and N^2 the set of customers. We represent the depot by 0 (source) and $n+1$ (sink) and extend the set N^1 by defining $N_0^1 = N^1 \cup \{0, n+1\}$. The set of nodes is defined as $N = N_0^1 \cup N^2$. The set $A^1 = \{(i,j) : i,j \in N_0^1, i \neq j, i \neq n+1, j \neq 0\}$ encompasses every arc that connects two parking locations or the depot and a parking location. The set $\widetilde{A}^2 = \{(i,j) : i,j \in N^2, i \neq j\}$ contains the arcs that connect every pair of customers. Let $(N^1 : N^2)$ represent the arcs that go from a node in N^1 to a node in N^2. We shall denote by $A^2 = \widetilde{A}^2 \cup (N^1 : N^2) \cup (N^2 : N^1)$ the set of arcs connecting two customers or a customer and a parking location. The set of arcs in the graph is denoted by $A = A^1 \cup A^2$.



(a) Instance. (b) Solution.

Figure 1: An illustrative example of the VRPTWMD-C2R.

A homogeneous fleet of vehicles with limits of capacity Q^1 and route duration T travels in the arcs of A^1. Each vehicle may carry from 1 to M_L deliverymen that serve the customers. Once the vehicle is parked, the deliverymen leave the vehicle to serve the customers. Both the vehicle fleet and the deliveryman crew are homogeneous and unlimited. We assume that the deliverymen travel with the same vehicle throughout the entire route. A vehicle remains parked while the deliverymen complete their tasks and waits for all of them to return before proceeding to the next cluster. Each deliveryman may perform at most one route per vehicle stop and has a capacity limit of Q^2. The deliverymen travel in the arcs of set A^2. We assume that each parking location has a limited transshipment capacity to reflect the fact that it is not viable to serve an indefinite amount of demand from a single parking location.

Figure 1 presents an example of the VRPTWMD-C2R. Figure 1a illustrates an instance of the problem, showing a depot, a set of customers, and a set of potential parking locations. In Figure 1b a feasible solution is portrayed. Only four out of the seven potential parking locations are effectively used, and the customers are clustered around these locations. The black arrows represent the vehicle routes, while the colored arrows within the clusters indicate the deliveryman routes. The vehicle on the right-hand side of the figure travels with two deliverymen. Upon arriving at the upper-right green cluster, the deliverymen leave the vehicle to serve the customers in parallel. Afterwards, they return to the vehicle and travel to the lower-right red cluster, where the process is repeated. Finally, the vehicle returns to the depot. The vehicle on the left-hand side of the figure travels with a single deliveryman, who serves all customers in the clusters visited by this vehicle.

Customers have time windows within which the service must start. Likewise, parking locations have time windows within which the deliverymen may leave the vehicle to start serving the customers. The vehicles and deliverymen can arrive earlier at the parking locations and customers and wait until the time window opening to start the service.

Every cluster is visited by exactly one vehicle and every customer by exactly one deliveryman. Every customer has a positive demand that must be completely fulfilled in a single visit of a deliveryman. The decisions of the problem are (i) which customers are to be assigned to each parking location, (ii) the number of vehicles to be used, (iii) the vehicle routes, (iv) the number of deliverymen traveling with each vehicle, and (v) the deliveryman routes.

4. Mathematical formulation

We introduce a novel mixed-integer programming compact formulation to formally define and solve the VRPTWMD-C2R. For clarity, variables and parameters associated with customers and deliveryman routes are identified with a superscript "2" (second echelon), while those related to vehicles and parking locations are denoted with a superscript "1" (first echelon). Also, nodes in N_0^1 are represented by i and j, and nodes in N^2 by h and k. We define $L = \{1, 2, \dots, M_L\}$ as the set of possible configurations (number) of deliverymen on a vehicle. The parameters included in the formulation are:

M_L Maximum number of deliverymen in each vehicle;

f^1, f^2 Fixed cost incurred from the use of a vehicle or deliveryman, respectively;

c^1, c^2 Unitary distance cost of vehicle and deliveryman routes, respectively;

Q^1, Q^2 Vehicle and deliveryman load capacity, respectively;

d_{ij}^1, d_{hk}^2 Distance between nodes i and $j, (i, j) \in A^1$, and h and $k, (h, k) \in A^2$ (asymmetrical);

t_{ij}^1, t_{hk}^2 Travel time between nodes i and $j, (i, j) \in A^1$, and h and $k, (h, k) \in A^2$ (asymmetrical);

s_h^2 Service time of customer $h \in N^2$;

s_i^1 Lower bound for the time spent in parking location $i \in N^1$ if this parking location is used. Defined as $s_i^1 = \min_{h \in N^2}\{t_{ih}^2 + s_h^2 + t_{hi}^2\},\ \forall\ i \in N^1$;

$[a_i^1, b_i^1], [a_h^2, b_h^2]$ Time window of parking location $i \in N^1$, and customer $h \in N^2$. We set the route duration limit of each vehicle, T, as b_{n+1}^1;

H_i Capacity of parking location $i \in N^1$;

q_h^2 Demand of customer $h \in N^2$;

The variables of the problem are:

x_{ijl}^1 Binary variable that indicates whether a vehicle travels from node i to node j with l deliverymen, $(i, j) \in A^1, l \in L$;

w_i^1 Arrival time at node $i \in N^1$;

w'^1_i Departure time from node $i \in N^1$;

u_i^1 Vehicle load after leaving node $i \in N^1$;

x_{hk}^2 Binary variable that indicates whether a deliveryman travels through arc $(h, k) \in A^2$;

w_h^2 Instant at which service at customer $h \in N^2$ begins;

u_h^2 Deliveryman load after leaving customer $h \in N^2$;

z_{jh} Binary variable that indicates whether customer $h \in N^2$ is served by a deliveryman traveling with a vehicle parked at parking location $j \in N^1$.

We present the following compact formulation (CF) to formally define the VRPTWMD-C2R:

$$\text{(CF)}\ \min \sum_{j\in N^1}\sum_{l\in L}(f^1+lf^2)x^1_{0jl} + c^1\sum_{(i,j)\in A^1}\sum_{l\in L} d^1_{ij}x^1_{ijl} + c^2\sum_{(h,k)\in A^2} d^2_{hk}x^2_{hk} \tag{1}$$

$$\text{s.t.}\ \sum_{i:(i,j)\in A^1}\sum_{l\in L} x^1_{ijl} \le 1,\ \forall\, j\in N^1 \tag{2}$$

$$\sum_{i:(i,j)\in A^1} x^1_{ijl} = \sum_{i:(j,i)\in A^1} x^1_{jil},\ \forall\, j\in N^1, l\in L \tag{3}$$

$$\sum_{i\in N^1} x^1_{0il} = \sum_{i\in N^1} x^1_{i(n+1)l},\ \forall\, l\in L \tag{4}$$

$$\sum_{h\in N^2} q^2_h z_{ih} \le H_i,\ \forall\, i\in N^1 \tag{5}$$

$$\sum_{j\in N^1} z_{jh} = 1,\ \forall\, h\in N^2 \tag{6}$$

$$\sum_{h:(h,k)\in A^2} x^2_{hk} = 1,\ \forall\, k\in N^2 \tag{7}$$

$$\sum_{h:(h,k)\in A^2} x^2_{hk} = \sum_{h:(k,h)\in A^2} x^2_{kh},\ \forall\, k\in N^2 \tag{8}$$

$$\sum_{h\in N^2} x^2_{ih} = \sum_{h\in N^2} x^2_{hi},\ \forall\, i\in N^1 \tag{9}$$

$$\sum_{k\in N^2} x^2_{jk} \le \sum_{i:(i,j)\in A^1}\sum_{l\in L} l x^1_{ijl},\ \forall\, j\in N^1 \tag{10}$$

$$x^2_{hk} + x^2_{kh} + z_{ih} - z_{ik} \le 1,\ \forall\, h,k\in N^2, h\neq k, i\in N^1 \tag{11}$$

$$x^2_{ih} \le z_{ih},\ \forall\, (i,h)\in (N^1:N^2) \tag{12}$$

$$x^2_{hi} \le z_{ih},\ \forall\, (h,i)\in (N^2:N^1) \tag{13}$$

$$w'^1_i \ge w^1_i,\ \forall\, i\in N^1 \tag{14}$$

$$w^1_j \ge w'^1_i + t^1_{ij} - M_{ij}\left(1-\sum_{l\in L} x^1_{ijl}\right),\ \forall\, i,j\in N^1, i\neq j \tag{15}$$

$$w^2_k \ge w^2_h + s^2_h + t^2_{hk} - M_{hk}(1-x^2_{hk}),\ \forall\, (h,k)\in \widetilde{A}^2 \tag{16}$$

$$w^2_k \ge w^1_i + t^2_{ik} - M_{ik}(1-x^2_{ik}),\ \forall\, (i,k)\in (N^1:N^2) \tag{17}$$

$$w'^1_i \ge w^2_h + s^2_h + t^2_{hi} - M_{hi}(1-x^2_{hi}),\ \forall\, (h,i)\in (N^2:N^1) \tag{18}$$

$$u^1_i \ge \sum_{h\in N^2} q^2_h z_{ih},\ \forall\, i\in N^1 \tag{19}$$

$$u^1_j \ge u^1_i + \sum_{h\in N^2} q^2_h z_{jh} - Q^1\left(1-\sum_{l\in L} x^1_{ijl}\right),\ \forall\, i,j\in N^1, i\neq j \tag{20}$$

$$u^2_k \ge u^2_h + q^2_k - Q^2(1-x^2_{hk}),\ \forall\, (h,k)\in \widetilde{A}^2 \tag{21}$$

$$x^1_{ijl}\in\{0,1\},\ \forall\, (i,j)\in A^1, l\in L \tag{22}$$

$$0\le u^1_i \le Q^1,\ \forall\, i\in N^1 \tag{23}$$

$$a^1_i \le w^1_i \le b^1_i,\ \forall\, i\in N^1 \tag{24}$$

$$a^1_i + s^1_i \le w'^1_i \le T - t^1_{i(n+1)},\ \forall\, i\in N^1 \tag{25}$$

$$x^2_{hk}\in\{0,1\},\ \forall\, (h,k)\in A^2 \tag{26}$$

$$a^2_h \le w^2_h \le b^2_h,\ \forall\, h\in N^2 \tag{27}$$

$$q_h^2 \leq u_h^2 \leq Q^2, \ \forall \ h \in N^2 \tag{28}$$

$$z_{jh} \in \{0, 1\}, \ \forall \ j \in N^1, h \in N^2. \tag{29}$$

The objective function (1) seeks to minimize fixed and variable costs of both vehicles and deliverymen. Constraints (2) limit the usage of each parking location to at most once. Constraints (3) and (4) ensure vehicle flow conservation. Constraints (5) limit the demand served by each parking location to its load capacity. Constraints (6) ensure that each customer is assigned to exactly one parking location and constraints (7) that every customer is visited exactly once. Constraints (8) and (9) are the deliveryman routes equivalent to (3) and (4). Constraints (10) limit the number of deliverymen leaving a parking location to serve the customers to the number of deliverymen that arrive at that parking location. Constraints (11) ensure that the deliverymen can only travel between nodes assigned to the same parking location. Constraints (12) and (13) define that a deliveryman can only travel between a parking location and a customer if this customer has been assigned to that parking location. Constraints (14) state that a vehicle can only leave a parking location after arriving at it. Constraints (15) and (16) define the time flow in vehicle and deliveryman routes, respectively. Constraints (17) and (18) synchronize vehicle and deliveryman routes. In these constraints, $M_{ij} = \max\{0, T - t^1_{i(n+1)} + t^1_{ij} - a^1_j\}$, $M_{hk} = \max\{0, b^2_h + s^2_h + t^2_{hk} - a^2_k\}$, $M_{ik} = \max\{0, b^1_i + t^2_{ik} - a^2_k\}$, and $M_{hi} = \max\{0, b^2_h + s^2_h + t^2_{hi} - a^1_i - s^1_i\}$. Constraints (19) define that the load of a vehicle after visiting a cluster is at least the sum of the demands assigned to the corresponding parking location. Constraints (20) and (21) control the load flow of vehicle and deliveryman routes, respectively. Constraints (22)–(29) define the variable domains.

4.1. Theoretical properties

In this section we present some theoretical properties of the problem and establish useful lower bounds to define our valid inequalities and solution methods. These results are formally defined and proved in Propositions 1 to 5 and Corollary 1.

Proposition 1. *If the triangular inequality holds, there is an optimal solution in which only parking locations with customers assigned to it are visited.*

Proof. Suppose that a vehicle visits nodes $i, j, k \in N^1$ in this sequence and that there is no customer assigned to parking location j. In this case, if the vehicle goes straight from node i to node k, the route is still feasible and the total distance is reduced by $d^1_{ij} + d^1_{jk} - d^1_{ik}$. Given that the triangular inequality holds, $d^1_{ij} + d^1_{jk} \geq d^1_{ik}$ and, hence, $d^1_{ij} + d^1_{jk} - d^1_{ik} \geq 0$. This "shortcut" leads to a vehicle route that is at most as costly as the previous one. Therefore, given a solution that visits a parking location without customers assigned to it, there is always a solution that is at least as good as this one and does not visit this parking location. $\square$

Corollary 1. *If the triangular inequality holds, there is an optimal solution in which a deliveryman leaves every parking location visited by a vehicle.*

Proposition 2. *If the triangular inequality holds, a lower bound on the time spent in a parking location $i \in N^1$ visited by l deliverymen is given by*

$$\sum_{h \in N^2} \left(s^2_h + (t^2_{ih} + t^2_{hi}) \frac{q^2_h}{Q^2} \right) \frac{z_{ih}}{l}.$$

Proof. Consider an instance of the asymmetric capacitated vehicle routing problem (ACVRP) with the depot represented by 0, the set of customers by N, the demands of a node $j \in N$ by q_j, and the travel time to and from the depot as t_{j0} and t_{0j}, respectively. Given this notation, one can show that the total travel time of the vehicles (summing up for all vehicles) in this instance is at least $\sum_{j\in N}(t_{0j}+t_{j0})\frac{q_j}{Q}$ by extending to the ACVRP the lower bound presented by Haimovich and Kan (1985) for the capacitated vehicle routing problem (CVRP) – following the logic presented in their paper.

In the VRPTWMD-C2R, once defined the customers assigned to a parking location, the dynamics of the deliverymen inside this cluster are similar to a vehicle routing problem with time windows (VRPTW) in which the parking location acts as the depot. Since the VRPTW is a more constrained version of the ACVRP, $\sum_{h\in N^2}(t^2_{ih}+t^2_{hi})\frac{q^2_h}{Q^2}z_{ih}$ is a lower bound on the total travel time inside cluster $i\in N^1$. The time spent in the cluster considers both the total travel time and the total service time. Also, if the cluster is visited by l deliverymen, in a best case scenario the total time is evenly divided between these deliverymen, yielding the lower bound presented above. □

Proposition 3. *If the triangular inequality holds, a lower bound on the total travel time of the vehicles is*

$$\frac{1}{Q^1}\sum_{i\in N^1}\sum_{h\in N^2}(t^1_{0i}+t^1_{i0})q^2_h z_{ih}.$$

Proof. Analogous to the proof of Proposition 2. □

Proposition 4. *If the triangular inequality holds, a lower bound on the total time the vehicles stay out of the depot is*

$$\frac{1}{Q^1}\sum_{i\in N^1}\sum_{h\in N^2}(t^1_{0i}+t^1_{i0})q^2_h z_{ih}+\frac{1}{M_L}\sum_{h\in N^2}\left(s_h+\frac{1}{Q^2}\sum_{i\in N^1}(t^2_{ih}+t^2_{hi})q^2_h z_{ih}\right).$$

Proof. By summing up the lower bound from Proposition 2 for all parking locations considering that they are visited by M_L deliverymen (resulting in the smallest possible lower bound) with the lower bound from Proposition 3 for the total travel time of the vehicles, one gets this lower bound. □

Proposition 5. *If the triangular inequality holds, a lower bound on the cost of the deliveryman routes inside a cluster $i\in N^1$ is*

$$c^2\sum_{h\in N^2}(d^2_{ih}+d^2_{hi})q^2_h z_{ih}.$$

Proof. Analogous to the proof of Proposition 2. □

4.2. Valid inequalities

With these results, the presented CF can be strengthened by the following valid inequalities (VIs):

$$\sum_{(i,j)\in A^1:i,j\in S}\sum_{l\in L}x^1_{ijl}\leq |S|-1,\ \forall\ S\subset N^1:|S|\in\{2,3\} \tag{30}$$

$$\sum_{(h,k)\in A^2:h,k\in S}x^2_{hk}\leq |S|-1,\ \forall\ S\subset N^2:|S|\in\{2,3\} \tag{31}$$

$$\sum_{j\in N^1}\sum_{l\in L}x^1_{0jl}\geq\left\lceil\frac{1}{Q^1}\sum_{h\in N^2}q^2_h\right\rceil \tag{32}$$

$$\sum_{\substack{(i,j)\in A^1\\ i\neq 0}} \sum_{l\in L} l x^1_{ijl} \geq \left\lceil \frac{1}{Q^2} \sum_{h\in N^2} q^2_h \right\rceil \tag{33}$$

$$x^1_{ijl} = 0,\ \forall\, i,j \in N^1, i\neq j, l\in L : a^1_i + s^1_i + t^1_{ij} > b^1_j \tag{34}$$

$$x^2_{hi} = 0, x^2_{ih} = 0, z_{ih} = 0,\ \forall\, i\in N^1, h\in N^2 : (a^1_i + t^2_{ih} > b^2_h) \vee (a^2_h + s^2_h + t^2_{hi} > T - t^1_{i(n+1)}) \tag{35}$$

$$x^2_{hk} = 0,\ \forall\, h,k\in N^2, h\neq k : (a^2_h + s^2_h + t^2_{hk} > b^2_k) \vee (q^2_h + q^2_k > Q^2) \tag{36}$$

$$\sum_{\substack{(i,j)\in A^1\\ i\neq 0}} \sum_{l\in L} x^1_{ijl} \geq P_{min} \tag{37}$$

$$x^2_{jk} \leq \sum_{i:(i,j)\in A^1} \sum_{l\in L} x^1_{ijl},\ \forall\, j\in N^1, k\in N^2 \tag{38}$$

$$\sum_{k\in N^2} x^2_{jk} \geq \sum_{i:(i,j)\in A^1} \sum_{l\in L} x^1_{ijl},\ \forall\, j\in N^1 \tag{39}$$

$$\sum_{h\in N^2} z_{jh} \geq \sum_{i:(i,j)\in A^1} \sum_{l\in L} x^1_{ijl},\ \forall\, j\in N^1 \tag{40}$$

$$z_{jh} \leq \sum_{i:(i,j)\in A^1} \sum_{l\in L} x^1_{ijl},\ \forall\, j\in N^1, h\in N^2 \tag{41}$$

$$\sum_{h\in N^2} z_{jh} \geq \sum_{h\in N^2} x^2_{jh},\ \forall\, j\in N^1 \tag{42}$$

$$z_{jh} \leq \sum_{k\in N^2} x^2_{jk},\ \forall\, j\in N^1, h\in N^2 \tag{43}$$

$$\sum_{h\in N^2} q^2_h z_{jh} \leq Q^2 \sum_{i:(i,j)\in A^1} \sum_{l\in L} l x^1_{ijl},\ \forall\, j\in N^1 \tag{44}$$

$$w'^1_i \geq a^1_i + s^1_i + (a^2_h + s^2_h + t^2_{hi} - a^1_i - s^1_i) z_{ih},\ \forall\, i\in N^1, h\in N^2 : a^2_h + s^2_h + t^2_{hi} > a^1_i + s^1_i \tag{45}$$

$$w^1_i \leq b^1_i + (b^2_h - t^2_{ih} - b^1_i) z_{ih},\ \forall\, i\in N^1, h\in N^2 : b^2_h - t^2_{ih} < b^1_i \tag{46}$$

$$w^2_h \geq a^2_h + (a^1_i + t^2_{ih} - a^2_h) z_{ih},\ \forall\, i\in N^1, h\in N^2 : a^1_i + t^2_{ih} > a^2_h \tag{47}$$

$$w^2_h \leq b^2_h + (T - t^1_{i(n+1)} - t^2_{hi} - s^2_h - b^2_h) z_{ih},\ \forall\, i\in N^1, h\in N^2 : T - t^1_{i(n+1)} - t^2_{hi} - s^2_h < b^2_h \tag{48}$$

$$w'^1_i - w^1_i \geq (t^2_{ih} + s^2_h + t^2_{hi}) z_{ih},\ \forall\, i\in N^1, h\in N^2 \tag{49}$$

$$w'^1_i - w^1_i \geq s^1_i,\ \forall\, i\in N^1 \tag{50}$$

$$Q^2 l (w'^1_i - w^1_i) \geq \sum_{h\in N^2} \left(Q^2 s^2_h + (t^2_{ih} + t^2_{hi}) q^2_h \right) z_{ih} - M_{il} \left(1 - \sum_{j:(i,j)\in A^1} \sum_{\bar{l}\in L:\bar{l}\leq l} x_{ij\bar{l}} \right),\ \forall\, i\in N^1, l\in L \tag{51}$$

$$T M_L Q^1 \sum_{j\in N^1} \sum_{l\in L} x^1_{0jl} \geq M_L \sum_{i\in N^1} \sum_{h\in N^2} (t^1_{0i} + t^1_{i0}) q^2_h z_{ih} + Q^1 \sum_{h\in N^2} \left(s^2_h + \frac{1}{Q^2} \sum_{i\in N^1} (t^2_{ih} + t^2_{hi}) q^2_h z_{ih} \right). \tag{52}$$

Constraints (30)–(37) are common in the literature (Dantzig et al., 1954; Ascheuer et al., 2001; Lysgaard et al., 2004; Yıldız et al., 2023), constraints (38)–(43) are adapted for the VRPTWMD-C2R from the valid inequalities proposed for the 2E-LRP by Senna et al. (2024b), and constraints (44)–(52) are novel valid inequalities proposed for this problem. Constraints (30) and (31) eliminate small subtours of two and three nodes in both vehicle and deliveryman routes. Constraints (32) define a lower bound on the number of vehicles used considering customer demands and vehicle capacity, and constraints (33) do the same for the deliverymen that leave the parking locations. Constraints (34)–(36) eliminate infeasible arcs and assignments due to time window incompatibility and deliveryman capacity. Constraint (37) defines that the number of parking locations visited is greater than a lower bound (P_{min}) on the number of parking locations needed to serve all customers considering their demands and the parking locations capacity. To define P_{min}, the parking locations should be ordered in a decreasing lexicographic order from the one with the largest to the one with the smallest capacity. The value of P_{min} is defined as the number of parking locations obtained by following this ordered list until the accumulated capacity is at least the sum of all customer demands. Constraints (38) state that deliverymen do not leave a parking location if it is not visited by a vehicle. Constraints (39) ensure that Corollary 1 holds. Constraints (40) guarantee that Proposition 1 holds. Constraints (41) state that no customer is assigned to a parking location if it is not visited by a vehicle. Constraints (42) define that deliverymen only leave a parking location if there are customers assigned to it. Constraints (43) ensure that no customer is assigned to a parking location if no deliveryman leaves it. Constraints (44) limit the total demand of the customers assigned to a parking location to the capacity of the deliverymen visiting this parking location. Constraints (45)–(48) define lower and upper bounds on the time variables based on the assignment of customers to parking locations. Constraints (49) state that the time spent in a parking location is at least the time to serve the customer that takes more time to be visited and served. Constraints (50) ensure that the time spent in a parking location $i \in N^1$ is greater than or equal to the lower bound s_i^1. Constraints (51) define the lower bound presented in Proposition 2 for the time spent in a parking location. In these constraints, $M_{il} = \sum_{h \in N^2}(Q^2 s_h^2 + (t_{ih}^2 + t_{hi}^2)q_h^2) - Q^2 l s_i^1$ is a sufficiently large number to ensure the validity of the constraints. Constraints (52) define a lower bound on the number of vehicles needed to serve the customers, based on the lower bound on the total time that the vehicles stay out of the depot from Proposition 4. On top of these VIs, time windows were tightened based on Ascheuer et al. (2001).

5. Benders decomposition

The VRPTWMD-C2R can be decomposed in a Benders fashion (Benders, 1962; Hooker and Ottosson, 2003) by reformulating the CF (1)–(29). Due to the high dependence of the deliveryman routes on clustering and vehicle routes, the master problem (MP) assigns customers to parking locations and defines the vehicle routes while the subproblem (SP) defines the deliveryman routes. To solve this reformulation of the VRPTWMD-C2R, we design a branch-and-Benders-cut (BBC) algorithm (Moreno et al., 2019, 2020; Senna et al., 2024a) that presents better performance than the CF, as indicated by the results of computational experiments discussed in Section 6.

To outline this reformulation, Section 5.1 presents the MP, Section 5.2 introduces the SP, Section 5.3 discusses the BBC, Section 5.4 proposes some improvements to the BBC, and Section 5.5 introduces a mixed-integer programming (MIP) heuristic for the problem that can be used to provide a good initial solution.

5.1. Master Problem

Let r represent a vehicle route that starts and ends at the depot, visiting a set of parking locations to which there are customers assigned to (r simultaneously represents the vehicle route and the customer assignment). Let $N_r^1 \subset N^1$ be the set of parking locations visited by this route, and $N_r^2 \subset N^2$ be the set of customers assigned to the parking locations in N_r^1. We shall represent the arcs of route r that are not connected to the depot as A_r^1. An assignment of customers to parking locations defines sets $N_r^{[i]}, i \in N^1$, that are the sets of customers assigned to the corresponding parking location $i \in N^1$. We shall refer to a set $N_r^{[i]}$ as a cluster.

Define R as the set of feasible pairs (r, l), where r represents a vehicle route and l denotes the number of deliverymen in this route. These pairs of parameters are all feasible with regard to constraints (2)–(29), since they represent vehicle routes and customer clustering that allow for feasible deliveryman routes. Given a pair (r, l), the cost of the deliveryman routes inside the clusters defined by $N_r^{[i]}, i \in N^1$, is c_{rl}. Let $\overline{R}$ be the set of infeasible pairs (r, l) when considering the deliveryman routes, i.e., the pairs that respect constraints (2)–(6), (14), (15), (19), (20), (22)–(25), and (29), but do not respect at least one of constraints (7)–(13), (16)–(18), (21), and (26)–(28). Finally, let $\eta_i, i \in N^1$, be a variable that represents the cost of the deliveryman routes inside cluster i.

With these definitions, the CF (1)–(29) can be reformulated as the following MP:

$$\text{(MP)} \min \sum_{j \in N^1} \sum_{l \in L} (f^1 + l f^2) x^1_{0jl} + c^1 \sum_{(i,j) \in A^1} \sum_{l \in L} d^1_{ij} x^1_{ijl} + \sum_{i \in N^1} \eta_i \tag{53}$$

$$\text{s.t. } (2)\text{–}(6), (14), (15), (19), (20), (22)\text{–}(25), (29)$$

$$\sum_{i \in N_r^1} \eta_i \geq c_{rl} \left(\sum_{(i,j) \in A_r^1} \sum_{\bar{l} \in L : \bar{l} \leq l} x^1_{ij\bar{l}} + \sum_{j \in N_r^1} \sum_{h \in N_r^{[j]}} z_{jh} - |A_r^1| - |N_r^2| + 1 \right), \quad \forall\, (r,l) \in R \tag{54}$$

$$\sum_{(i,j) \in A_r^1} \sum_{\bar{l} \in L : \bar{l} \leq l} x^1_{ij\bar{l}} + \sum_{j \in N_r^1} \sum_{h \in N_r^{[j]}} z_{jh} \leq |A_r^1| + |N_r^2| - 1, \ \forall\, (r,l) \in \overline{R}. \tag{55}$$

The objective function (53) is equivalent to (1) with the cost of the deliveryman routes calculated based on variables η_i. Constraints (54) and (55) are optimality and feasibility cuts based on path-cuts (Parada et al., 2024; Senna et al., 2024a). We shall refer to the MP without the optimality and feasibility cuts as the relaxed MP (RMP).

To define VIs, let $R_{il} \subset R$ be the set of all assignments of customers to parking location i that are feasible considering the deliveryman routes when visited by $l \in L$ deliverymen in a back-and-forth trip from the depot (a vehicle route that only visits parking location $i \in N^1$). Accordingly, let $\overline{R}_{il} \subset \overline{R}$ be the set of all assignments of customers to parking location i that creates clusters that are infeasible when visited by $l \in L$ deliverymen in back-and-forth trips from the depot.

The MP can be strengthened by VIs (30), (32), (34), (35), (37), (40), (41), (44)–(46), (49)–(52). We also propose the following valid inequalities:

$$Q^2 \eta_i \geq c^2 \sum_{h \in N^2} (d^2_{ih} + d^2_{hi}) q^2_h z_{ih}, \ \forall\, i \in N^1 \tag{56}$$

$$\eta_i \geq c^2 (d^2_{ih} + d^2_{hi}) z_{ih}, \ \forall\, i \in N^1, h \in N^2 \tag{57}$$

$$\eta_i \geq c_{rM_L} \left(\sum_{h \in N_r^{[i]}} z_{ih} - |N_r^{[i]}| + 1 \right), \ \forall\, i \in N^1, (r, M_L) \in R_{iM_L} \tag{58}$$

$$\sum_{h \in N_r^{[i]}} z_{ih} + \sum_{j : (i,j) \in A^1} \sum_{\bar{l} \in L : \bar{l} \leq l} x^1_{ij\bar{l}} \leq |N_r^{[i]}|, \ \forall\, i \in N^1, (r,l) \in \overline{R}_{il} \tag{59}$$

$$w'^1_i - w^1_i \geq t_{rl} \left(\sum_{h \in N_r^{[i]}} z_{ih} + \sum_{j:(i,j) \in A^1} \sum_{\bar{l} \in L: \bar{l} \leq l} x^1_{ij\bar{l}} - |N_r^{[i]}| \right), \ \forall\ i \in N^1, (r,l) \in R_{il}. \tag{60}$$

Constraints (56) impose the lower bound presented in Proposition 5 for the cost of deliveryman routes inside a cluster. Constraints (57) state that the cost of the deliveryman routes associated to a parking location is at least the cost of visiting the farthest customer associated to it. Constraints (58) provide a lower bound on the cost of the deliveryman routes in a cluster. Constraints (59) eliminate infeasible assignments. Constraints (60) define a lower bound (t_{rl}) on the time spent in each parking location by the vehicle visiting it depending on the customers assigned to it and the number of deliverymen on the vehicle. In constraints (59) and (60), when $l = M_L$, the summation in x_{ijl} may be replaced by 1, since this is the best case scenario for costs and feasibility.

The optimality and feasibility cuts (54) and (55) and the VIs (58)–(60) are of exponential cardinality. Therefore, it is impractical to enumerate all of them a priori. Instead, one declares the RMP and starts to solve it in a branch-and-cut scheme. When a solution is found, the cuts and VIs needed for this solution are separated and included in the model. This leads to the BBC algorithm described in Section 5.3. The separation of cuts and VIs is made by solving the SP described next.

5.2. Subproblem

Given a pair (r,l), we define an SP that is separable by vehicle route. To simplify notation, we shall represent $N_r^{[i]}$ by $N^{[i]}$ in this context. Also, parking location i will be represented by nodes 0_i and n_i+1 for deliveryman routes source and sink, respectively, with $n_i = |N^{[i]}|$. Let $N_0^{[i]} = N^{[i]} \cup \{0_i, n_i+1\}$. We define the complete directed graph $G^{[i]} = (N_0^{[i]}, A^{[i]})$, in which $A^{[i]} = \{(h,k) \in \widetilde{A}^2 : h,k \in N^{[i]}\} \cup (\{0_i\} : N^{[i]}) \cup (N^{[i]} : \{n_i+1\})$.

The SP is given by

$$\text{(SP)} \ \min\ c^2 \sum_{i \in N_r^1} \sum_{(h,k) \in A^{[i]}} d^2_{hk} x^2_{hk} \tag{61}$$

$$\text{s.t.} \quad \sum_{h:(h,k) \in A^{[i]}} x^2_{hk} = 1, \ \forall\ k \in N^{[i]}, i \in N_r^1 \tag{62}$$

$$\sum_{h:(h,k) \in A^{[i]}} x^2_{hk} = \sum_{h:(k,h) \in A^{[i]}} x^2_{kh}, \ \forall\ k \in N^{[i]}, i \in N_r^1 \tag{63}$$

$$\sum_{h \in N^{[i]}} x^2_{0_i h} = \sum_{h \in N^{[i]}} x^2_{h(n_i+1)}, \ \forall\ i \in N_r^1 \tag{64}$$

$$\sum_{h \in N^{[i]}} x^2_{0_i h} \leq l, \ \forall\ i \in N_r^1 \tag{65}$$

$$w^2_k \geq w^2_h + s^2_h + t^2_{hk} - M_{hk}(1 - x^2_{hk}), \ \forall\ (h,k) \in A^{[i]}, i \in N_r^1 \tag{66}$$

$$w^2_{0_j} \geq w^2_{n_i+1} + t^1_{ij}, \ \forall\ (i,j) \in A_r^1 \tag{67}$$

$$u^2_k \geq u^2_h + q^2_k - Q^2(1 - x^2_{hk}), \ \forall\ (h,k) \in A^{[i]}, i \in N_r^1 \tag{68}$$

$$x^2_{hk} \in \{0,1\}, \ \forall\ (h,k) \in A^{[i]}, i \in N_r^1 \tag{69}$$

$$a^2_h \leq w^2_h \leq b^2_h, \ \forall\ h \in N_0^{[i]}, i \in N_r^1 \tag{70}$$

$$q^2_h \leq u^2_h \leq Q^2, \ \forall\ h \in N^{[i]}, i \in N_r^1. \tag{71}$$

The objective function (61) minimizes the cost of the deliveryman routes inside the clusters visited by route r. Constraints (62)–(64) are equivalent to (7)–(9) but restricted to the customers visited in the route. Constraints (65) limit the number of deliveryman routes in each cluster to the number of deliverymen traveling in the corresponding vehicle. Constraints (66) control the time flow of the deliveryman routes. Constraints (67) control the time flow along the vehicle route. Constraints (68) control the load flow inside the clusters. Constraints (69)–(71) define variable domains.

The SP can be strengthened by the following VIs:

$$\sum_{(h,k)\in A^{[i]}:h,k\in S} x^2_{hk} \leq |S| - 1,\ \forall\, i \in N^1_r, S \subset N^{[i]} : |S| \in \{2,3\} \tag{72}$$

$$x^2_{hk} = 0,\ \forall\, i \in N^1_r, h, k \in N^{[i]}, h \neq k : (a^2_h + s^2_h + t^2_{hk} > b^2_k) \vee (q^2_h + q^2_k > Q^2) \tag{73}$$

$$w^2_{n_i+1} - w^2_{0_i} \geq \max\left\{\frac{1}{l}\sum_{h\in N^{[i]}}\left[s^2_h + (t^2_{0_ih} + t^2_{h(n_i+1)})\frac{q^2_h}{Q^2}\right], \max_{h\in N^{[i]}}\left\{t^2_{0_ih} + s^2_h + t^2_{h(n_i+1)}\right\}\right\},\quad \forall\, i \in N^1_r. \tag{74}$$

Constraints (72) and (73) are equivalent to constraints (31) and (36) but restricted to the customers visited by the vehicle route. Constraints (74) define a lower bound on the time spent in each cluster as the maximum of the lower bound discussed in Proposition 2 and the time needed to serve the most time-consuming customer. Time windows are also tightened based on Ascheuer et al. (2001). Since in the SP the vehicle route is already defined, this tightening becomes very efficient.

The SP is used to define the optimality and feasibility cuts (54) and (55). When the SP is feasible, the value of the objective function for an optimal solution is used to define the parameter c_{rl} of constraints (54) for the pair $(r, l) \in R$. If the SP is not feasible, the pair (r, l) belongs to $\overline{R}$ and, hence, a feasibility cut (55) must be added to the MP.

The SP is also used to define VIs (58)–(60). To separate VIs (58), one defines the SP based on a vehicle route that goes from the depot to a parking location and back to the depot with M_L deliverymen. For VIs (59), the procedure is the same, but with the number of deliverymen that are actually traveling in the vehicle that visits the corresponding cluster (l). Finally, VIs (60) are separated by replacing the objective function (61) with $w'^1_i - w^1_i$ and solving the SP for a route that goes back and forth from the depot to customer $i \in N^1$ and considering the number of deliverymen $l \in L$ in the vehicle. The objective function value of an optimal solution to this problem is used to define the value of parameter t_{rl} of constraints (60).

5.3. Branch-and-Benders-cut algorithm

Due to the exponential nature of the cuts (54) and (55) and the VIs (58)–(60), it is impractical to enumerate all of them to solve the VRPTWMD-C2R. Instead, we solve the problem in a branch-and-Benders-cut scheme (Moreno et al., 2019, 2020; Senna et al., 2024a). To this extent, the RMP strengthened by the polynomial VIs is solved in a branch-and-cut fashion. Every time an integer solution is found, the SP is solved to separate the optimality and feasibility cuts (54) and (55) and VIs (58)–(60). The following steps summarize the BBC algorithm:

1. Declare the RMP with the polynomial VIs (30), (32), (34), (35), (37), (40), (41), (44)–(46), (49)–(52), (56), and (57) and start the branch-and-cut algorithm;

2. Every time a feasible integer solution is found, separate VIs (58)–(60) by solving the SP restricted to a single cluster for all clusters in the current solution. Separate also the feasibility and

optimality cuts (54) and (55) by solving the SP defined for all pairs (r, l) of the current solution. Add the separated cuts and VIs to the model;

3. If the current solution is feasible given the deliveryman routes, compute the overall solution cost by including the cost update given by the SP. If this cost is lower than that of the incumbent solution, update the incumbent;

4. Continue the branch-and-cut solution procedure by proceeding to the next node in the branch-and-cut tree. If a new feasible integer solution is found, return to step 2. If the time limit is reached or the optimality gap reaches the optimality tolerance, interrupt the algorithm procedure.

This algorithm can be implemented in modern general-purpose MIP solvers by means of callbacks.

5.4. Improvements

For some instances, the separation procedures of the BBC may take a few seconds for each route, which leads to a long time spent in separation procedures, i.e., solving the MIPs that correspond to the SP. This leads to a reduction in the rate at which the branch-and-cut nodes are processed. Although essential to the BBC algorithm, it would be better to separate these cuts only when they are needed and the corresponding SP is useful, i.e., it corresponds to an important route or assignment. Moreover, at the beginning of the solution procedure, the lower bound is too low and the first solutions found by the solver are usually of poor quality. Therefore, it would be interesting to separate cuts when solutions are better and the lower bound is not so low.

To overcome these issues, we propose a two-phase BBC (2P-BBC). In the first phase, only one cut is separated, enough to cut off the solution presented by the solver while reducing the computational burden of separating every possible cut. In the second phase, every VI (58)–(60) and cut (54)–(55) is separated. The second phase starts upon reaching a gap plateau, i.e., when the solution procedure remains a long time without significantly improving the optimality gap, which indicates that both the lower bound and the upper bound found by the BBC have not significantly improved. The following steps are executed in the procedure of the 2P-BBC:

1. Declare the RMP with the polynomial VIs (30), (32), (34), (35), (37), (40), (41), (44)–(46), (49)–(52), (56), and (57) and start the branch-and-cut algorithm;

2. Every time a feasible integer solution is found, verify whether a gap plateau has been reached. If so, go to step 5;

3. Start to separate VIs (58)–(60) by solving the SP restricted to a single cluster. Upon finding a VI that cuts off the current solution, include this VI in the model and go to step 7 without separating other VIs;

4. Start to separate cuts (54) and (55), one route at a time. Upon finding a cut that cuts off the current solution, include this cut in the model and go to step 7. If no cut has been found, go to step 6;

5. Separate all VIs (58)–(60) by solving the SP restricted to a single cluster and all feasibility and optimality cuts (54) and (55) by solving the SP defined by the vehicle routes and customer assignments of the solution;

6. If the current solution is feasible given the deliveryman routes and its cost is lower than that of the incumbent solution after computing the cost of the deliveryman routes, update the incumbent;

7. Continue the branch-and-cut algorithm by proceeding to the next node in the branch-and-cut tree. If a new feasible integer solution is found, return to step 2. If the time limit is reached or the optimality gap reaches the optimality tolerance, interrupt the algorithm procedure.

Like the algorithm in Section 5.3, this can be implemented in general-purpose MIP solvers by means of callbacks. When implementing this algorithm, it is important to be careful in step 3 to ensure that, in successive callback calls, different parking locations are selected for VI separation. Otherwise, several VIs would be separated for a single node (e.g., the one with smallest index), having many cuts related to this parking location and none related to the others. This deteriorates the algorithm's performance and significantly increases the number of included constraints, most of them non-binding in an optimal solution. In our implementation, we have ordered the parking locations and defined that the first parking location to be processed in a callback call is the subsequent of the one that had a cut separated in the previous call. If no VI was separated for this parking location, the next node would be analyzed until one VI was found or it was proved that there was no VI (58)–(60) that cuts off the current solution. This way, if a VI was included for a parking location in a callback call, it would be the last one to be analyzed in the next call.

5.5. MIP heuristic

For some instances, the CF and the BBC showed to be slow in finding good feasible solutions. Thus, providing good initial solutions lead to better overall performance of the algorithm. This is specially important for the two-phase BBC, since the delayed separation of VIs and cuts makes it more difficult for the algorithm to update the incumbent solution at the beginning of the solution procedure.

To overcome this issue, we have developed a MIP heuristic that finds a good feasible solution in a short amount of time. The procedure is based on defining, for each customer, a list of the parking locations that have a time window opening that varies at most $0.1T$ from the moment that the customer's time window opens. The heuristic consists in solving the CF (with VIs) by limiting the parking locations to which each customer can be assigned to the α closest ones from this list. The resulting MIP is then solved by a general-purpose MIP solver for a few minutes or until it finds a solution with optimality gap within a tolerance. This solution is then used as a MIP start for the BBC (or the CF). If by constraining the assignment of each customer we obtain an infeasible problem, the heuristic is solved iteratively by increasing α by one until it finds a feasible solution for the problem.

6. Computational experiments

Computational experiments were performed to evaluate the suitability of the proposed methodology and to obtain managerial insights on the problem. All algorithms were implemented in C++ and use Gurobi 11.0 solver. The optimality gap tolerance was set at 10^{-7}, the time limit at 3,600s, and the memory limit at 32GB. The experiments were performed on computers equipped with 2xAMD Rome 7532 processors running at 2.46GHz and using eight threads. For the MIP heuristic, the optimality tolerance was set at 10%, and the time limit at 300s; the initial value of α was three.

In Section 6.1, the instances used in the experiments are presented. Section 6.2 evaluates the performance of the CF and VIs for solving the problem with the general-purpose MIP solver and

Section 6.3 does the same for the BBC. In Section 6.4, the algorithms are further evaluated by analyzing their convergence. In Section 6.5, managerial insights are presented, shedding light onto the importance of considering both the customers clustering and the deliveryman routes in the problem. Finally, Section 6.6 makes a sensitivity analysis by evaluating the impact of some deliveryman characteristics on the solution. All instances and detailed results are available at `https://www.dep.ufscar.br/munari/vrptwmd/`.

6.1. Instances

Our experiments were based on the instance set proposed by Senna et al. (2024a) for the VRPTWMD with two-level routing with 50 nodes (10 parking locations and 40 customers). These instances were generated by the authors based on the Solomon instances for the VRPTW (Solomon, 1987), having the first ten nodes of the original instance representing a parking location and randomly generating customers around them. These instances have predefined clusters, but they are ignored for the VRPTWMD-C2R.

Following Senna et al. (2024a), we have considered the cost parameters to be $(f^1, c^1, f^2, c^2) = (1000, 10, 100, 1)$ and that the deliverymen travel at one third of the vehicle speed. A limit of $M_L = 3$ deliverymen in each vehicle was considered. Distances were calculated based on the euclidean distance truncated to integers. We ran the Floyd-Warshall algorithm (Cormen et al., 2009) on these distances to ensure the triangular inequality was valid. Travel times were processed accordingly. In all instances, we considered that the deliveryman capacity is 50, since it is the largest individual demand in the Solomon instances (Solomon, 1987).

6.2. Compact formulation

We first assess the performance of the MIP solver with CF (1)–(29) and different sets of VIs. Five different configurations were compared. The first one (CF1) corresponds to CF without VIs. The second (CF2) represents CF with VIs from the literature (30)–(37). The third configuration (CF3) has CF with VIs from the literature (30)–(37) and the new ones that are related to the binary variables and load constraints (38)–(44). Configuration CF4 includes CF and all VIs (30)–(52): the ones included in the other configurations and the ones related to time variables. We also evaluate the impact of using the MIP heuristic to provide a MIP start to CF4, referred to as CF4H.

The performance of the MIP solver with different CF configurations varies significantly depending on the instance class (C, R, or RC) to which the original Solomon instance belongs. When generating instances for the VRPTW, Solomon proposed three different classes of instances. Class “C” has its nodes separated in clusters, class “R” has the nodes uniformly randomly generated, and, for class “RC”, some of the nodes were generated as in class “C” and some as in class “R” (Solomon, 1987). Therefore, with the generation of new customers around these nodes for the VRPTWMD (Senna et al., 2024a), the geographical distribution of customers and parking locations varies significantly for different instance classes. In class “C”, there are many parking locations close to each other, having many possibly interesting parking locations to which the customers can be assigned. On the contrary, in class “R”, parking locations are usually far apart, having few candidate parking locations that are interesting for each customer. Class “RC” has an intermediate behavior.

These results are summarized in Table 1. The results are presented divided by instance class and aggregated by all instances as well. In these tables, for each CF configuration, “LR” represents the optimal value of the linear programming relaxation of the VRPTWMD-C2R. “LB” and “UB” stand, respectively, for the lower and upper bounds reported by the MIP solver at the end of the solving procedure. "Gap (%)" corresponds to the optimality gap and "Time (s)" to the runtime in seconds. All these values represent the average for all instances in the corresponding classes. Moreover, "# optimals" and "# no feasible solution" indicate, respectively, the number of instances for which the MIP solver could prove optimality for the best solution found and could not find any feasible solution. For some instance classes, the solver could not find a feasible solution for all instances. In these cases, the corresponding values of UB and gap were reported as "N/A". Detailed results are available as supplementary material and also at `https://www.dep.ufscar.br/munari/vrptwmd/`.

Class	Metric	CF1	CF2	CF3	CF4	CF4H
C (9 instances)	LR	69	1,710	1,716	1,737	1,737
	LB	754	1,773	1,910	3,307	3,329
	UB	N/A	5,085	5,102	4,612	3,612
	Gap (%)	N/A	64.93	61.66	26.85	7.85
	Time (s)	3,600	3,600	3,600	3,600	3,601
	# optimals	0	0	0	0	0
	# no feasible solution	2	0	0	0	0
R (12 instances)	LR	176	3,801	3,804	3,877	3,877
	LB	3,239	5,730	5,791	8,253	8,236
	UB	N/A	8,714	8,707	8,551	8,551
	Gap (%)	N/A	35.33	34.75	3.65	3.80
	Time (s)	3,300	3,009	3,015	2,217	1,878
	# optimals	1	2	2	5	6
	# no feasible solution	3	0	0	0	0
RC (8 instances)	LR	101	3,955	3,962	3,964	3,964
	LB	1,128	4,232	4,250	5,363	5,509
	UB	N/A	9,646	9,883	8,928	8,490
	Gap (%)	N/A	55.66	56.69	39.05	34.94
	Time (s)	3,601	3,600	3,601	3,600	3,601
	# optimals	0	0	0	0	0
	# no feasible solution	2	0	0	0	0
All (29 instances)	LR	122	3,195	3,200	3,236	3,236
	LB	1,885	4,089	4,162	5,921	5,961
	UB	N/A	7,845	7,913	7,433	7,001
	Gap (%)	N/A	50.13	49.16	20.62	13.65
	Time (s)	3,476	3,356	3,358	3,028	2,888
	# optimals	1	2	2	5	6
	# no feasible solution	7	0	0	0	0

Table 1: Results of the CF configurations.

Regarding the LR, all results indicate that the linear programming relaxation of CF1 is very weak. The inclusion of VIs from the literature (CF2) leads to average values of LR that are over 25 times higher than the ones presented by CF1. The other VIs, however, do not impact much the value of LR, representing around 1% increase in the average value from CF2 to CF4. The greater differences are observed for instances of class R, suggesting that the new VIs affect more instances with more spread customers than those with customers closer to parking locations.

Comparing CF1 and CF2 with respect to the performance of the solver while optimizing the corresponding MIP problem, the difference in the strength of the LR has a significant impact on the algorithm's performance, since with CF2 all instances have a feasible solution found, whereas with CF1 the solver cannot find any feasible solution for seven instances. Moving on to CF3, the solver performance is overall improved, despite the differences between CF2 and CF3 being discrete. The greatest impact is for instances of class C, for which there is a gap improvement of 3.27%.

CF4 presents a great improvement in the results. On average, the LB increases 42.26% compared to the CF3 value, which, combined with a 6.07% UB improvement, leads to a 28.54% gap improvement.

Three new optimal solutions were also found for instances of class R. Moreover, the inclusion of the heuristic solution as a MIP start for the MIP solver provides major improvements in the solver performance, mainly for the instances of class C. In fact, for these instances, the UB is decreased by 21.68%, leading to a 19.00% gap reduction. For instances of class R, one extra optimal solution is found, but the average LB is worse for CF4H than for CF4. For instances of class RC, there are improvements in the LB and UB that lead to a 4.11% gap improvement. The overall average gap is reduced by 6.97%.

These results show the positive impact encompassed by the proposed valid inequalities and the use of the MIP heuristic to provide a MIP start. The solver under configuration CF1 (without VIs) has a poor performance due to its very weak LR. In fact, it cannot even find a feasible solution for 24.14% of the instances. When including all VIs and the heuristic (CF4H), the LB is increased by 216.23%. Moreover, five new optimal solutions are found and all instances have a feasible solution found, in contrast with the seven ones for which the solver could not find any solution under CF1.

6.3. Branch-and-Benders-cut

In this section, we compare the performance of different configurations of the BBC algorithm. Since our experiments with VIs show that all of them are beneficial for the solver performance, we have included all presented VIs in the BBC. More specifically, VIs (30), (32), (34), (35), (37), (40), (41), (44)–(46), (49)–(52), (56), and (57) were included in the MP, and VIs (72)–(74) in the SP. The first configuration is BBC1, which represents the BBC with all polynomial VIs but without exponential VIs (58)–(60). BBC2 corresponds to BBC1 with the inclusion of VIs (58)–(60). BBC2H includes the MIP heuristic solution as a MIP start for BBC2 . Finally, 2P-BBC2H has also the two-phase scheme discussed in Section 5.4. The results are presented in Table 2. This table also includes the results for CF4 and CF4H for comparison, since they were the ones with the best performance among the CF configurations.

Comparing BBC1 with BBC2, BBC2 presents better results on average, although their performance differ based on the instance class. For instances of class R, they have equivalent behaviors, finding the optimal solution for all instances and greatly outperforming the CFs. For those of class C, BBC2 outperforms BBC1, having a 3.20% smaller gap average. For class RC, the difference is the greatest one, since BBC1 is unable to find a feasible solution for one instance while BBC2 finds a feasible solution for all of them.

Upon the inclusion of the MIP heuristic (BBC2H), there is a great improvement in the average performance. For instances of class C, the UB is reduced by 25.40% compared to that of BBC2, leading to a 19.26% improvement in the average gap. For instances of class RC, the behavior is similar, with a 15.68% gap reduction. On average, the UB of BBC2H is 10.89% lower than that of BBC2, and the gap is improved by 10.31%. There is also one extra optimal solution found. These results suggest that the BBCs without the heuristic have trouble finding good initial solutions for some instances, as discussed in Section 5.5.

With the two-phase scheme, 2P-BBC2H leads to an additional 0.33% gap improvement compared to BBC2H. For instances of class C, there is an LB improvement that leads to a 0.11% gap reduction compared to BBC2H. For class RC, the LB is improved by 0.64%, the UB by 1.85%, and the gap by 1.10%.

The comparison of 2P-BBC2H with CF4H shows that 2P-BBC2H outperforms the solver with the CFs. In fact, the average LB of 2P-BBC2H is 2.37% higher than that of CF4H and the UB is

Class	Metric	CF4	CF4H	BBC1	BBC2	BBC2H	2P-BBC2H
C (9 instances)	LB	3,307	3,329	3,240	3,260	3,269	3,275
	UB	4,612	3,612	4,902	4,854	3,621	3,624
	Gap (%)	26.85	7.85	32.23	29.03	9.77	9.66
	Time (s)	3,600	3,601	3,605	3,604	3,605	3,604
	# optimals	0	0	0	0	0	0
	# no feasible solution	0	0	0	0	0	0
R (12 instances)	LB	8,253	8,236	8,539	8,539	8,539	8,539
	UB	8,551	8,551	8,539	8,539	8,539	8,539
	Gap (%)	3.65	3.80	0.00	0.00	0.00	0.00
	Time (s)	2,217	1,878	68	204	251	276
	# optimals	5	6	12	12	12	12
	# no feasible solution	0	0	0	0	0	0
RC (8 instances)	LB	5,363	5,509	5,280	5,060	5,592	5,628
	UB	8,928	8,490	N/A	10,060	8,360	8,205
	Gap (%)	39.05	34.94	N/A	48.53	32.85	31.75
	Time (s)	3,600	3,601	3,604	3,605	3,597	3,573
	# optimals	0	0	0	0	1	1
	# no feasible solution	0	0	1	0	0	0
All (29 instances)	LB	5,921	5,961	5,995	5,941	6,091	6,102
	UB	7,433	7,001	N/A	7,815	6,964	6,921
	Gap (%)	20.62	13.65	N/A	22.40	12.09	11.76
	Time (s)	3,028	2,888	2,141	2,198	2,215	2,218
	# optimals	5	6	12	12	13	13
	# no feasible solution	0	0	1	0	0	0

Table 2: Results of the BBCs.

1.14% lower, leading to a 1.89% gap reduction. Moreover, the average runtime is 23.20% shorter for 2P-BBC2H, which finds an optimal solution for seven extra instances (a 116.67% increase compared to CF4H).

All these results show that both the proposed VIs and the BBCs significantly improve the performance of the MIP solver, since CF1 (without VIs) presents a very poor performance. The proposed VIs, lower bounds, and heuristic lead to a much better performance while maintaining the formulation compact (i.e., with a polynomial number of variables and constraints). The Benders decomposition (BBCs) leads to a very good performance, despite generating a formulation with an exponential number of constraints. 2P-BBC2H clearly has the best performance among all developed methods.

To further understand the behavior of the BBC algorithms, Table 3 presents details of the cut separation procedures of the different BBC approaches. In these tables, "# optimality cuts (54)" and "# feasibility cuts (55)" respectively indicate the average number of optimality and feasibility cuts included in these instances. Likewise, "# VIs (58)", "# VIs (59)", and # VIs (60)" correspond to the average number of exponential VIs that were included by the BBCs. Finally, "Separation time (s)" indicates the average time spent in separation procedures and cut inclusion (the time spent in the solver callback).

This table shows a massive reduction in the number of optimality (54) and feasibility (55) cuts needed when including the exponential VIs (58), (59), and (60). On average, the total number of cuts (optimality and feasibility together), from BBC1 to BBC2, is reduced by 93.77%. The greater difference is for instances of class RC, with the number of cuts going from 11,707 in BBC1 to 696 in BBC2. Most of these cuts are optimality cuts. The drawback of the inclusion of these VIs is an increase in the separation time, which is of 38.22%. However, as documented in Table 2, this difference clearly pays off since BBC2 outperforms BBC1.

Class	Metric	BBC1	BBC2	BBC2H	2P-BBC2H
C (9 instances)	# optimality cuts (54)	4,276	589	775	780
	# feasibility cuts (55)	17,551	770	689	545
	# VIs (58)	–	2,831	1,941	1,629
	# VIs (59)	–	205	41	39
	# VIs (60)	–	2,001	1,507	1,312
	Separation time (s)	2,430	2,609	2,907	2,812
R (12 instances)	# optimality cuts (54)	69	18	6	4
	# feasibility cuts (55)	141	4	3	0
	# VIs (58)	–	34	17	17
	# VIs (59)	–	10	3	3
	# VIs (60)	–	31	20	18
	Separation time (s)	12	183	95	126
RC (8 instances)	# optimality cuts (54)	2,706	564	371	196
	# feasibility cuts (55)	12,905	397	251	113
	# VIs (58)	–	2,336	1,310	1,229
	# VIs (59)	–	798	302	267
	# VIs (60)	–	2,031	1,087	1,156
	Separation time (s)	1,896	3,213	3,117	2,921
All (29 instances)	# optimality cuts (54)	2,102	346	345	298
	# feasibility cuts (55)	9,065	350	284	200
	# VIs (58)	–	1,537	971	851
	# VIs (59)	–	288	97	87
	# VIs (60)	–	1,194	776	734
	Separation time (s)	1,282	1,772	1,801	1,731

Table 3: Algorithmic details of the BBCs.

The inclusion of the MIP heuristic solution as a MIP start reduces even further the average number of cuts and VIs that are included, since the algorithm starts from a better incumbent solution than the BBC without the initial solution. This way, there are fewer solutions of low quality that are evaluated in the separation procedures, leading to the separation of only more interesting cuts. Indeed, there is a reduction of 9.63% in the number of cuts and of 38.92% in the number of VIs. The greatest reduction is for instances of class RC, in which the average number of included VIs goes from 5,165 in BBC2 to 2,699 in BBC2H (52.26% reduction). Although these VIs are useful to improve the algorithm performance, as clearly shown by the fact that BBC2 outperforms BBC1, their exponential nature makes them excessively numerous. Therefore, ideally, only a small set of useful VIs should be included in the model. Hence, this reduction in their number is overall beneficial. Combined with the quality of the initial solution, this leads to a much better performance of BBC2H compared to BBC2, as discussed above. Finally, the number of included cuts and VIs is further reduced by the two-phase approach. The reduction in the number of cuts is of 20.83%, while in the number of VIs is of 9.33%.

Another interesting result from Table 3 is the inverse relation between the number of included cuts and VIs and the performance of the algorithm. This has been discussed above for the performance comparison of different algorithms when applied to the same instances. Nevertheless, this can also be seen by comparing the performance of the algorithms across different instance classes. The BBCs present the best performance for instances of class R, finding an optimal solution for all instances. For this class, the average number of cuts is lower than five for all BBCs, with a low number of VIs included. The separation times are also of less than 200s. This suggests that, for these instances, the RMP finds good solutions and the proposed lower bounds (discussed in Section 4.1) work properly. The good quality of the lower bounds is mainly due to the fact that this instance class has the parking locations far apart and, hence, assigning customers to parking locations that are not the closest ones leads to high values of the proposed lower bounds, helping the RMP to find good solutions for the VRPTWMD-C2R.

For instances of classes C and RC, the average gap is not negligible. Therefore, the number of included cuts and VIs represents the amount of cuts that could be separated given the time limit, not the number of cuts effectively needed to find an optimal solution. Considering that the algorithms' performances are better for instances of class C, the separated cuts and VIs were more useful for this instance class than for class RC. Together with the instances characteristics, this suggests that instances of class C have many interesting and similar solutions due to the high number of potentially interesting clusters. Moreover, these solutions are similar because the parking locations are clustered together. However, they find it difficult to prove optimality due to the many different solutions that cannot be differentiated by the RMP with the proposed lower bounds, since these bounds are weak when distances are small.

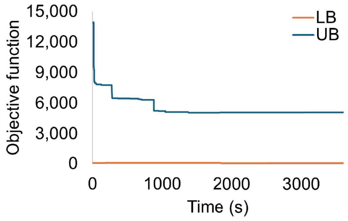
Instances of class RC have characteristics that are intermediary between classes C and R, leading to the combination of the high impact of the distance between different parking locations that is characteristic of class R with the high number of potentially good clusters that degrades the performance for instances of class C. This leads to instances that are very hard to solve since there are many potential clusters but they have very different costs since some of the parking locations are close together and others are far apart. The time needed to serve each cluster also varies significantly for the same reasons. If in class C the BBC does not close the gap because there are many potential solutions, but the LB quality is overall good because these solutions are very similar, in class RC, the BBC also has trouble closing the gap due to the number of solutions, but LB quality is overall poor because these solutions can be very different.

6.4. Convergence

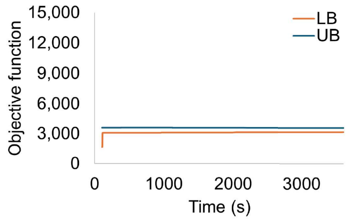
To better understand how the different algorithms compare regarding the solving procedure, we now look into their convergence curves. Figures 2 to 4 illustrate how the LB and UB evolve throughout the runtime for three different approaches: the solver upon solving CF1 (CF without VIs), the solver with CF4H (the best performing CF), and the 2P-BBC2H (the best performing BBC). These graphs are presented for instances C102, R104, and RC105, which represent common behaviors among the instances of their classes.

Figure 2a illustrates what is shown in Table 1: the LR provided by CF1 is very weak, and the solver is not able to increase it with the branch-and-cut procedure, being under 100 during over 3,600s for an instance whose best known LB is 3,149. Likewise, the UB starts from a very high value and, although it significantly decreases, it remains much higher than the one found by the other two approaches, leaving a gap of 98.40% after 3,600s of runtime. For this instance (and most of those of class C), CF4H and 2P-BBC2H have similar behaviors. As portrayed in Figures 2b and 2c, both start a little later, since the heuristic takes a few seconds, but they start from an incumbent solution that is much better than the one found by CF1 after one hour. Likewise, the LB increases fast, being over 3,000 for both approaches before 500s of runtime. Nevertheless, there is little improvement either in the LB or in the UB afterwards, and we observe the tailing off effect on the results of both approaches.

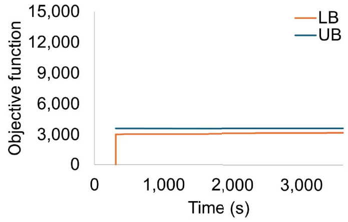
For instance R104, however, the behaviors are different. As indicated in Figure 3a, CF1 starts from a high UB and low LB that see no improvement past 500s of runtime. The presence of VIs and the heuristic solution greatly improve the solver performance, as shown in Figure 3b. CF4H already starts from much better LB and UB compared to CF1, and these values are improved until 1,000s. Nonetheless, after this point the solver presents a tailing off effect and remains with a gap of over 19%



(a) CF1.



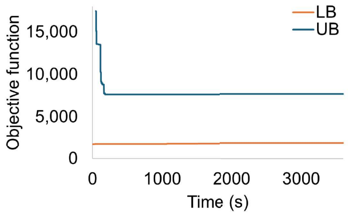
(b) CF4H.



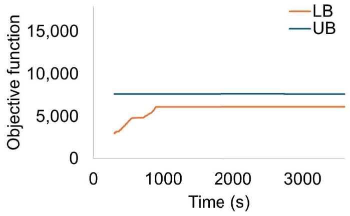
(c) 2P-BBC2H.

Figure 2: Convergence curves for instance C102.

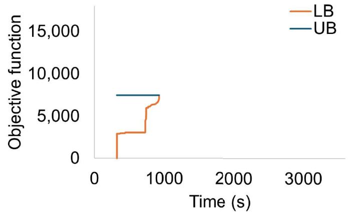
until the end of the solving procedure. 2P-BBC2H greatly differs from CF4H. Figure 3c indicates that this algorithm is able to converge to an optimal solution in under 1,000s. The ability to overcome the tailing off effect presented by CF4H and actually converge to the optimal solution value is the reason why the BBCs greatly outperform the solver with the CFs in instance class R.



(a) CF1.



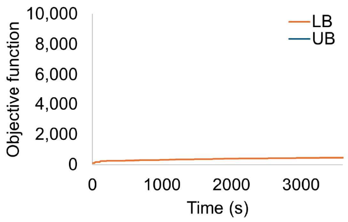
(b) CF4H.



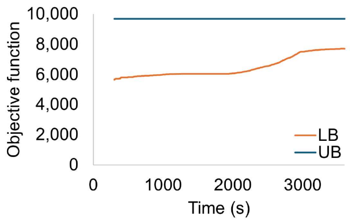
(c) 2P-BBC2H.

Figure 3: Convergence curves for instance R104.

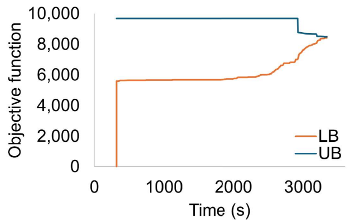
Instance RC105 contrasts with the other two for the solver with CF1 is unable to find a feasible solution. This is represented in Figure 4a by the absence of the UB curve. Moreover, the LB remains at a very low value. The inclusion of the heuristic provides an initial solution for the CF4H and the 2P-BBC2H. Figure 4b shows that the inclusion of VIs significantly increase the LR in CF4H, with the LB starting from over 5,000, in contrast with CF1 that, after 3,600s remains with an LB of under 1,000. This LB is improved throughout the solving procedure, reaching 7,719. The UB, nevertheless, is not improved by the solver branch-and-cut, leading to a remaining gap of 20.32% by the end of the runtime. 2P-BBC2H, however, is able to keep improving the LB and, after around 3,000s, the UB starts to decrease, as shown in Figure 4c. This leads to the algorithm finding the optimal solution value.



(a) CF1.



(b) CF4H.



(c) 2P-BBC2H.

Figure 4: Convergence curves for instance RC105.

These experiments confirm the fact that CF1 has a very poor LR and leads to poor solver performance. The MIP heuristic provides good initial solutions that significantly improve the solution quality. Moreover, the proposed VIs allow the solver to drastically increase the LB compared to the formulation without VIs. Despite these improvements, the solver with CF4H has a tendency to a tailing off effect, being unable to reduce the optimality gap. 2P-BBC2H represents an improvement in this behavior, overcoming the tailing off effect in many instances. This leads to over the double of instances with proved optimality, and better average LB, UB, and gap compared to CF4H.

6.5. The impact of optimizing clustering and deliveryman routes

As discussed in Section 2, the VRPTWMD was proposed by Pureza et al. (2012) with two simplifying hypotheses: (*i*) the customer clusters can be predefined, and (*ii*) the deliveryman routes can be preprocessed. Senarclens de Grancy and Reimann (2015) and Senarclens de Grancy (2015) extended the problem by relaxing hypothesis (*i*), while keeping hypothesis (*ii*), resulting in the VRPTWMD with customer clustering (VRPTWMD-C). Senna et al. (2024a) also extended the VRPTWMD by relaxing hypothesis (*ii*), while keeping hypothesis (*i*), creating the VRPTWMD with two-level routing (VRPTWMD-2R). Note that the VRPTWMD-C2R is the first to relax both of these hypotheses.

In the previous sections, we have discussed the performance of the proposed methodologies to solve the VRPTWMD-C2R. It is important, however, to assess the relevance of including the decisions of clustering and deliveryman routes in the problem, i.e., the impact on the solution quality of the VRPTWMD-C2R compared to the other VRPTWMD variants mentioned above. To this extent, we have performed some experiments simulating the different variants.

To ensure the optimal solution would be found for every variant, we have generated smaller instances, with five potential parking locations and 20 customers (hereinafter referred to as size 5–20), following what was proposed by Senna et al. (2024a). In their paper, they randomly generate the customers' coordinates based on a normal distribution with average at the parking locations' coordinates and standard deviation $\sigma = 3$. To further evaluate the impact of clustering, we have extended our analysis by generating instances with varying customer dispersion. To this extent, we have followed the procedure proposed by Senna et al. (2024a) and discussed in Section 6.1, but generated instances with different values of standard deviation for the customers coordinates ($\sigma \in \{1, 3, 5\}$). This way, there are instances with the customers closer to ($\sigma = 1$) and farther from ($\sigma = 5$) the parking locations.

As discussed in Section 6.1, the instances used in these experiments have predefined clusters that were ignored in the VRPTWMD-C and the VRPTWMD-C2R. Nevertheless, these clusters were used to simulate the VRPTWMD and the VRPTWMD-2R, by setting the variables z_{ih} to match the assignment provided by the instance. Furthermore, to simulate the preprocessing of deliveryman routes, we have assumed that, instead of making direct trips between customers, in both the VRPTWMD and the VRPTWMD-C, the deliveryman must always come back to the vehicle to take more goods as discussed in Section 2. This can be done by redefining the distances (travel times) between customers as the distances (travel times) passing through the parking location instead of the euclidean distances.

The results of these experiments are presented in Table 4 for a subset of instances to which it was possible to prove optimality for all configurations (with either CF4H or 2P-BBC2H). In this table, on top of presenting the results by instance class and standard deviation σ of customer's coordinates, the aggregated total is also provided. "Cost" represents the overall solution cost, "# veh." corresponds to the number of vehicles used in the solution, "# del." indicates the size of the deliveryman crew used, "veh. dist." stands for the distance traveled by the vehicles, and "del. dist." shows the distance traveled by the deliverymen. All presented values are computed as averages.

The results clearly indicate that, the higher the σ, the higher the difference between the solutions of the variants and, hence, the greater the importance of including the clustering and the deliveryman routes in the optimization. As expected, the costs of the VRPTWMD are the highest, since it is

Class	σ	Metric	VRPTWMD	VRPTWMD-C	VRPTWMD-2R	VRPTWMD-C2R
C	1	Cost	2,057	2,049	2,039	2,027
		# veh.	1.33	1.33	1.33	1.33
		# del.	1.33	1.33	1.33	1.33
		Veh. dist.	54.00	52.67	54.00	52.67
		Del. dist.	50.67	55.33	32.33	34.00
	3	Cost	2,225	2,168	2,171	2,089
		# veh.	1.33	1.33	1.33	1.33
		# del.	2.00	1.67	2.00	1.33
		Veh. dist.	53.67	53.00	53.67	53.67
		Del. dist.	154.67	138.00	100.67	86.00
	5	Cost	2,351	2,297	2,209	2,173
		# veh.	1.33	1.33	1.33	1.33
		# del.	2.33	2.00	1.67	1.67
		Veh. dist.	54.00	54.00	54.00	53.67
		Del. dist.	244.00	224.00	169.00	136.67
R	1	Cost	3,732	3,732	3,672	3,672
		# veh.	2.00	2.00	2.00	2.00
		# del.	2.33	2.33	2.00	2.00
		Veh. dist.	144.33	144.33	144.00	144.00
		Del. dist.	55.33	55.33	32.33	32.33
	3	Cost	3,952	3,952	3,803	3,803
		# veh.	2.00	2.00	2.00	2.00
		# del.	4.00	4.00	3.00	3.00
		Veh. dist.	141.67	141.67	141.67	141.67
		Del. dist.	135.33	135.33	86.33	86.33
	5	Cost	5,569	5,569	4,891	4,891
		# veh.	3.00	3.00	2.67	2.67
		# del.	7.67	7.67	5.00	5.00
		Veh. dist.	156.67	156.67	155.67	155.67
		Del. dist.	236.00	236.00	168.00	168.00
RC	1	Cost	2,114	2,112	2,028	2,027
		# veh.	1.00	1.00	1.00	1.00
		# del.	2.00	2.00	1.33	1.33
		Veh. dist.	86.33	86.33	86.33	86.33
		Del. dist.	50.67	48.67	31.67	30.67
	3	Cost	4,109	4,102	3,972	3,964
		# veh.	2.00	2.00	2.00	2.00
		# del.	4.33	4.00	3.67	3.33
		Veh. dist.	153.67	157.00	151.00	154.33
		Del. dist.	138.67	132.00	95.00	87.67
	5	Cost	6,151	4,934	5,312	4,176
		# veh.	3.00	2.33	2.67	2.00
		# del.	6.67	6.00	4.67	4.33
		Veh. dist.	225.33	178.00	201.33	159.00
		Del. dist.	231.33	220.67	165.67	153.00
Total		Cost	3,584	3,435	3,344	3,203
		# veh.	1.89	1.81	1.81	1.74
		# del.	3.63	3.44	2.74	2.59
		Veh. dist.	118.85	113.74	115.74	111.22
		Del. dist.	144.07	138.37	97.89	90.52

Table 4: The impact of different VRPTWMD variants on the solution quality.

the most constrained variant, and those of the VRPTWMD-C2R are the lowest, since it generalizes all others. In general, the costs of VRPTWMD-2R are lower than those of the VRPTWMD-C. Furthermore, the distance traveled by the deliverymen is always smaller for the variants that optimize the deliveryman routes than for those that consider them to be computed a priori.

For instances of class C, an interesting result is that the vehicle distances are the same for the VRPTWMD and the VRPTWMD-2R, suggesting that the vehicle routes do not change. Looking in more detail, for instances with $\sigma = 1$, the difference in the solutions is mostly concentrated in the distance traveled by the deliverymen, with little impact in the overall cost. For those with $\sigma = 3$ and $\sigma = 5$, however, differences in the number of deliverymen also appear, although the size of the vehicle fleet is always the same. As a consequence, the deliveryman routes distance is significantly impacted, having a 43.99% reduction from the VRPTWMD to the VRPTWMD-C2R in the instances with $\sigma = 5$, leading to a 7.57% cost reduction. The cost improvement of the VRPTWMD-C2R compared to the VRPTWMD-C is 5.40% and to the VRPTWMD-2R is 1.63% for those instances.

For instances of class R, the solutions of the VRPTWMD and the VRPTWMD-C are always the same. Accordingly, the solutions of the VRPTWMD-2R and the VRPTWMD-C2R are also equivalent. This suggests that, for these instances, the clustering can be easily preprocessed with little impact on the solution. This happens because these instances have their parking locations very far apart and, hence, clustering becomes trivial. The impact of considering the deliveryman routes, however, is not negligible. In fact, this leads to a reduction on the size of the deliveryman crew that ranges from 14.16% for instances with $\sigma = 1$ to 34.81% for those with $\sigma = 5$. Accordingly, the reduction on the distance traveled by the deliverymen ranges from 28.81% for instances with $\sigma = 5$ to 41.57% for those with $\sigma = 1$. Combined with some reductions in the vehicle fleet size and the distance traveled by the vehicles, this leads to a cost reduction that goes from 1.61% for instances with $\sigma = 1$ to 12.17% for those with $\sigma = 5$.

For instances of class RC, the number of vehicles used is the same for all variants and instances with $\sigma = 1$ and $\sigma = 3$. For instances with $\sigma = 5$, however, the reduction from the VRPTWMD to the VRPTWMD-C2R is of 33.33%. For instances with $\sigma = 1$, the main difference among the solutions of the different variants is on the distance traveled by the deliverymen, which is significantly reduced by the variants that optimize these routes. This leads to a reduction on the average solution cost of 4.12% from the VRPTWMD to the VRPTWMD-C2R. The clustering, however, has little impact. The behavior for $\sigma = 3$ is similar. Nonetheless, for instances with $\sigma = 5$, the clustering has a huge impact. In fact, the cost reduction from the VRPTWMD to the VRPTWMD-C is of 19.79% while from the VRPTWMD-2R to the VRPTWMD-C2R it is of 21.39%. Combined with the cost reduction obtained by the inclusion of the deliveryman routes in the optimization, this leads to an average solution cost of the VRPTWMD-C2R that is 32.11% smaller than that of the VRPTWMD.

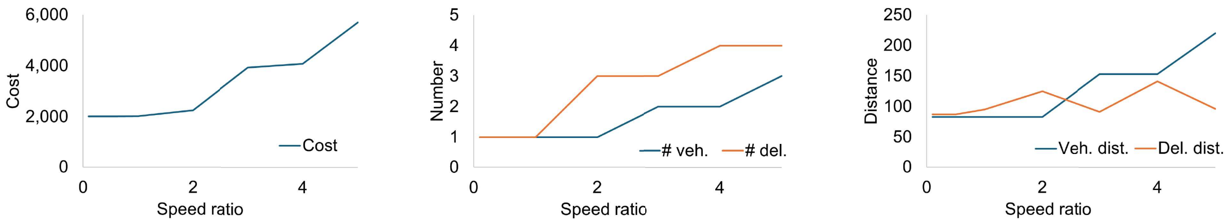
Considering the overall average of instances, it is clear that the optimization of the deliveryman routes is of major relevance in the optimal size of the deliveryman crew and the distance traveled by them. Moreover, clustering is important depending on the instances characteristics, as expected. On the one hand, for instances that have parking locations far apart from each other and customers close to them, the clustering becomes trivial and, hence, its optimization is not impactful. On the other hand, for instances that have many interesting candidate parking locations for each customer, clustering becomes relevant and its optimization may lead to major cost reductions. On the overall average, the VRPTWMD-C2R has a solution cost that is 10.63% lower than that of the VRPTWMD.

6.6. Sensitivity analysis

To conclude the managerial insights, we evaluate the impact of two essential deliveryman characteristics in the solution: speed and capacity. Figure 5 illustrates the impact of changing the speed ratio of the vehicles and deliverymen for instance RC104 of size 5–20 (all presented solutions were proved optimal by either the solver with CF4H or the 2P-BBC2H). This is the ratio between the vehicles speed and the deliverymen speed (the higher the ratio, the slower the deliverymen). Figure 5a shows the impact of the speed on the total solution cost. It shows that there is little cost variation if the deliverymen are faster than the vehicles or twice as slow. However, when the speed ratio starts to increase, the total cost is greatly affected, being two times higher for deliverymen that travel at one third of the vehicles speed than for those that travel at half the vehicle speed. When the speed ratio is five, the total cost is three times higher than when it is two.

This cost difference is caused by the impact on the number and distance traveled by vehicles and deliverymen. Figure 5b shows how the number of vehicles and deliverymen change in the optimal solution for this instance depending on the speed ratio. If the deliverymen travel at the same speed of the vehicles, or higher, the optimal solution has only one vehicle and one deliveryman. As the speed ratio increases, these numbers also rise, with the number of deliverymen reaching four and the number of vehicles reaching three. The behavior for the distance traveled by vehicles and deliverymen is different, as portrayed in Figure 5c. It clearly shows that, as the speed ratio increases, there is a tendency to increase the distance traveled by the vehicles. Nevertheless, the distance traveled by the deliverymen increases when the speed ratio increases but the number of vehicles remains the same. As the number of vehicles increase, the distance traveled by the deliverymen is actually reduced. This behavior is coherent with real applications, since deliverymen usually cannot travel much by foot, being impossible to increase much the distance traveled by them. As their speed reduces, it is needed either to increase their number or to have more vehicles to serve the same customers.

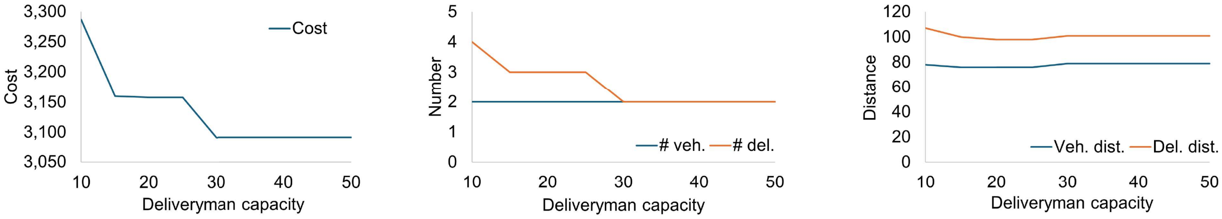
These results shed light onto the efficiency of alternative delivery schemes. As the speed of the deliveryman increases, the costs are reduced. This could be implemented in practice by changing walking carriers by bicycles, motorcycles, or drones, for example.



(a) Total cost. (b) Fleet and crew sizes. (c) Distance.

Figure 5: The impact of the deliveryman speed on the solution of instance RC104 of size 5–20.

Another analysis is made regarding the deliverymen capacity. Figure 6 illustrates this for instance C106 of size 5–20 (all presented solutions were proved optimal by either the solver with CF4H or the 2P-BBC2H). As expected, the higher the deliveryman capacity, the lower the total costs, as shown in Figure 6a. In Figure 6b, the impact of this capacity on the number of vehicles and deliverymen in the optimal solution is portrayed. Unlike the speed, the deliveryman capacity does not affect the number of vehicles needed. Regarding the deliverymen, on the one hand, their number can be reduced by half by increasing their capacity, since they can perform longer and more efficient routes. On the other hand, as shown in Figure 6c, the total distance traveled by them is not significantly affected. Even though increasing their capacity allows for reducing the number of deliverymen needed, they end up traveling farther, creating little impact on the total distance. In practice, the deliveryman capacity can be increased by having them using small carts that would allow them to carry more packages. It is clear from Figure 6 that this scheme creates an opportunity for cost reduction.



(a) Total cost. (b) Fleet and crew sizes. (c) Distance.

Figure 6: The impact of the deliveryman capacity on the solution of instance C106 of size 5–20.

7. Conclusion

In this paper, we have introduced a variant of the vehicle routing problem with time windows and multiple deliverymen (VRPTWMD). This problem emulates a last-mile delivery scheme that has vehicles traveling with more than one deliveryman to increase the number of customers that can be served with a single stop of the vehicle while reducing the overall time that the vehicles stay parked throughout the route. Since deliveryman costs and greenhouse gas emissions are usually smaller than those of the vehicles, this allows for a cheaper and greener delivery system.

As originally proposed, the VRPTWMD preprocesses the customers to be served by each stop of the vehicles (clustering) and the deliveryman routes inside the clusters. In previous studies, the problem had been extended to encompass either the clustering or the deliveryman routes in the optimization, but never both. With this paper, we have bridged this gap by introducing the vehicle routing problem with time windows, multiple deliverymen, customer clustering, and two-level routing (VRPTWMD-C2R), which is a two-echelon vehicle routing problem with applications in last-mile delivery.

We have formally defined the VRPTWMD-C2R by means of a mathematical formulation. Theoretical properties and lower bounds have been discussed and used to propose valid inequalities. The problem has also been decomposed in a Benders fashion to develop a branch-and-Benders-cut algorithm to solve it. Computational experiments show the suitability of the proposed methodology to solve the problem.

Furthermore, managerial insights were provided to shed light onto the importance of optimizing the customer clustering and the deliveryman routes simultaneously. Our results show that the optimization of deliveryman routes is always beneficial. The clustering, however, depends on the instance characteristics. In fact, if customers are closely distributed around parking locations that are far apart from each other, clustering becomes trivial and its optimization is not relevant. Nevertheless, in situations that have many parking locations close to each other with customers distributed around them without an obvious clustering pattern, optimizing clustering is of major relevance.

Finally, potential directions for future work include exploring variants that allow the deliverymen to return to the vehicles at different parking locations or switch vehicles during the route. Other interesting extensions would be the study of the problem under uncertainties (e.g., in the demand or travel time) by means of robust or stochastic optimization. New methods based on metaheuristics could also provide better solutions for large-scale instances.

Acknowledgements

This work was supported by the São Paulo Research Foundation (FAPESP) [grant numbers 2013/07375-0, 2021/14441-5, 2022/09679-5, 2022/05803-3], the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) [Finance Code 001], the National Council for Scientific and Technological Development (CNPq) [grant number 405702/2021-3, 304618/2023-3, 314079/2023-8], and the Canadian Natural Sciences and Engineering Research Council (NSERC) [grant number 2019-00094]. This support is greatly acknowledged. We thank the Digital Research Alliance of Canada for providing high-performance parallel computing facilities.

References

Alfandari, L., Ljubić, I., and De Melo da Silva, M. (2022). A tailored Benders decomposition approach for last-mile delivery with autonomous robots. *European Journal of Operational Research*, 299(2):510–525.

Álvarez, A. and Munari, P. (2016). Metaheuristic approaches for the vehicle routing problem with time windows and multiple deliverymen. *Gestão e Produção*, 23(2):279–293.

Álvarez, A. and Munari, P. (2017). An exact hybrid method for the vehicle routing problem with time windows and multiple deliverymen. *Computers & Operations Research*, 83:1–12.

Amine Masmoudi, M., Mancini, S., Baldacci, R., and Kuo, Y.-H. (2022). Vehicle routing problems with drones equipped with multi-package payload compartments. *Transportation Research Part E: Logistics and Transportation Review*, 164:102757.

Ascheuer, N., Fischetti, M., and Grötschel, M. (2001). Solving the asymmetric travelling salesman problem with time windows by branch-and-cut. *Mathematical Programming*, 90:475–506.

Bayliss, C., Bektaş, T., Tjon-Soei-Len, V., and Rohner, R. (2023). Designing a multi-modal and variable-echelon delivery system for last-mile logistics. *European Journal of Operational Research*, 307(2):645–662.

Bektaş, T. and Laporte, G. (2011). The pollution-routing problem. *Transportation Research Part B: Methodological*, 45(8):1232–1250.

Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252.

Cabrera, N., Cordeau, J.-F., and Mendoza, J. E. (2022). The doubly open park-and-loop routing problem. *Computers & Operations Research*, 143:105761.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). *Introduction to Algorithms*. The MIT Press, Boston, USA, 4 edition.

Cuda, R., Guastaroba, G., and Speranza, M. G. (2015). A survey on two-echelon routing problems. *Computers & Operations Research*, 55:185–199.

Dantzig, G., Fulkerson, R., and Johnson, S. (1954). Solution of a large-scale traveling-salesman problem. *Journal of the Operations Research Society of America*, 2(4):393–410.

De La Vega, J., Munari, P., and Morabito, R. (2019). Robust optimization for the vehicle routing problem with multiple deliverymen. *Central European Journal of Operations Research*, 27(4):905–936.

De La Vega, J., Munari, P., and Morabito, R. (2020). Exact approaches to the robust vehicle routing problem with time windows and multiple deliverymen. *Computers & Operations Research*, 124:105062.

Haimovich, M. and Kan, A. H. G. R. (1985). Bounds and heuristics for capacitated routing problems. *Mathematics of Operations Research*, 10(4):527–542.

Hooker, J. N. and Ottosson, G. (2003). Logic-based Benders decomposition. *Mathematical Programming*, 96(1):33–60.

Lysgaard, J., Letchford, A. N., and Eglese, R. W. (2004). A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, 100:423–445.

Martinez-Sykora, A., Lamas-Fernandez, C., Bektaş, T., Cherrett, T., and Allen, J. (2020). Optimised solutions to the last-mile delivery problem in london using a combination of walking and driving. *Annals of Operations Research*, 295:645–693.

Moreno, A., Munari, P., and Alem, D. (2019). A branch-and-Benders-cut algorithm for the crew scheduling and routing problem in road restoration. *European Journal of Operational Research*, 275(1):16–34.

Moreno, A., Munari, P., and Alem, D. (2020). Decomposition-based algorithms for the crew scheduling and routing problem in road restoration. *Computers & Operations Research*, 119:104935.

Munari, P. and Morabito, R. (2018). A branch-price-and-cut algorithm for the vehicle routing problem with time windows and multiple deliverymen. *TOP*, 26(3):437–464.

Parada, L., Legault, R., Côté, J.-F., and Gendreau, M. (2024). A disaggregated integer L-shaped method for stochastic vehicle routing problems with monotonic recourse. *European Journal of Operational Research*, 318(2):520–533.

Pureza, V., Morabito, R., and Reimann, M. (2012). Vehicle routing with multiple deliverymen: Modeling and heuristic approaches for the VRPTW. *European Journal of Operational Research*, 218(3):636–647.

Senarclens de Grancy, G. (2015). An adaptive metaheuristic for vehicle routing problems with time windows and multiple service workers. *Journal of Universal Computer Science*, 21(9):1143–1167.

Senarclens de Grancy, G. and Reimann, M. (2015). Evaluating two new heuristics for constructing customer clusters in a VRPTW with multiple service workers. *Central European Journal of Operations Research*, 23(2):479–500.

Senna, F., Coelho, L. C., Morabito, R., and Munari, P. (2024a). An exact method for a last-mile delivery routing problem with multiple deliverymen. *European Journal of Operational Research*, 317(2):550–562.

Senna, F., Coelho, L. C., Morabito, R., and Munari, P. (2024b). The two-echelon location-routing problem: A comparative analysis of novel and existing compact formulations. Technical report CIRRELT-2024-15, Université de Montréal.

Sluijk, N., Florio, A. M., Kinable, J., Dellaert, N., and Van Woensel, T. (2023). Two-echelon vehicle routing problems: A literature review. *European Journal of Operational Research*, 304(3):865–886.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254–265.

Souza Neto, J. F. and Pureza, V. (2016). Modeling and solving a rich vehicle routing problem for the delivery of goods in urban areas. *Pesquisa Operacional*, 36(3):421–446.

Yıldız, E. A., İsmail Karaoğlan, and Altiparmak, F. (2023). An exact algorithm for two-echelon location-routing problem with simultaneous pickup and delivery. *Expert Systems with Applications*, 231:120598.