LI Simpósio Brasileiro de Pesquisa Operacional
Limeira–SP, 02 a 06 de setembro de 2019

The Heterogeneous Vehicle Routing Problem with Multiple Deliverymen and Simultaneous Pickup and Delivery

Matheus Augusto Fernandes de Assunção
Instituto de Ciencias Matemáticas e de Computação – Universidade de São Paulo – Brasil
Avenida Trabalhador São-carlense, 400 - Centro- São Carlos
matheusfernandes@usp.br

Maristela Oliveira dos Santos
Instituto de Ciencias Matemáticas e de Computação – Universidade de São Paulo – Brasil
Avenida Trabalhador São-carlense, 400 - Centro- São Carlos
mari@icmc.usp.br

ABSTRACT

In this article, the Heterogeneous Vehicle Routing Problem with Multiple Deliverymen and Simultaneous Pickup and Delivery (HVRPMDSPD) is discussed. This problem is inspired by real world reverse logistics applications where a bottling company, using a fleet of different vehicles, must deliver beverages to clients and also pick up empty cases of bottles to recycle. Moreover, it is usually not possible to stop at every client's location, and many clients might be close to one another. This proximity allows the vehicles to park at certain areas from where the deliverymen can serve the clients nearby on foot. Employing more deliverymen on a given route cuts down service times, but also potentially raises operational costs. The HVRPMDSPD is modelled mathematically as a Mixed Integer Linear Programming Model (MILP) problem. A constructive heuristic was developed so as to provide an initial solution for the solver CPLEX, and the solutions obtained were investigated.

KEYWORDS. Heterogeneous Vehicle Routing. Simultaneous Pickup and Delivery. Mathematical Model. Constructive Heuristic.

Paper topics (L&T - Logística e Transportes)

1. Introduction

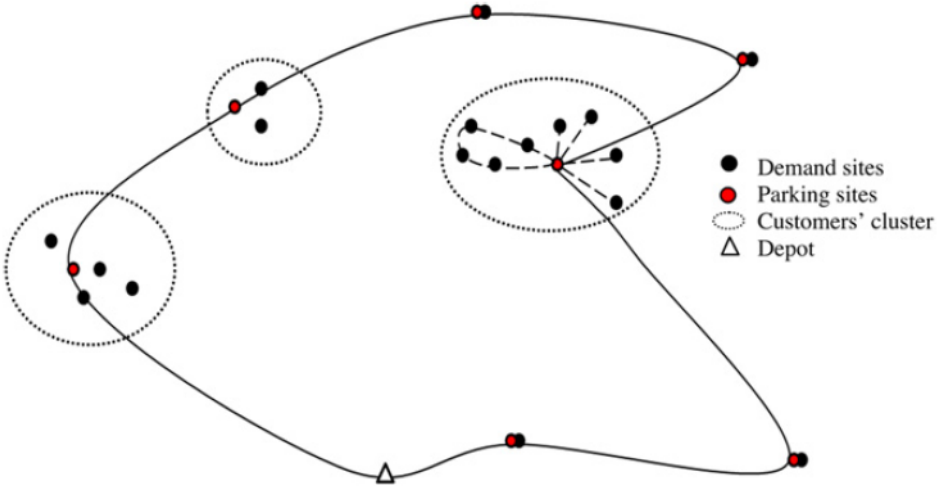
The Vehicle Routing Problem (VRP) is an extension of the Traveling Salesman Problem (TSP), where a salesman must find the shortest route that passes through every town he must visit. The Vehicle Routing Problem can be defined as a decision problem where a fleet of vehicles must deliver goods to clients while also traveling the shortest routes possible, starting and ending at a depot. The goal can be to minimize operational costs, distance traveled, fleet size, operational time, or even to maximize the number of clients served by the vehicles. Other constraints can also be added to the problem, such as time windows (where a client can only be served within a specific time period), maximum route time duration (for instance, an 8 hour shift), simultaneous pickup and delivery, number of clients in each route, or even the order in which clients must be visited [Braekers et al., 2016].

The VRPTWMD, or Vehicle Routing Problem with Time Windows and Multiple Deliverymen, was first described in Pureza and Morabito [2010]. The VRPTWMD is a variation of the VRP inspired by a real world application where a bottling company had to deliver drinks to a number of clients in a dense city environment. The company's trucks, instead of stopping at each and every client, would park near some clients and perform the deliveries to those clients on foot. This was important because it might not always be possible to park at every clients's location, due to traffic and maneuverability. This meant that employing more deliverymen could impact delivery times significantly. In that article, the authors demonstrate the advantage of incorporating multiple deliverymen by altering the instances from Solomon [1987] and solving them using a Tabu Search heuristic. The authors justify this approach with the real world example of a bottling company in São Paulo, explaining that the company must adhere to the following planning constraints:

- availability of the vehicles in the fleet for each delivery period;
- maximum capacity of the available vehicles;
- stock of each of the products requested;
- time availability for the delivery of each product;
- delivery priority of each client;
- time windows in one or more clients.

In Pureza et al. [2012], an MILP model for the VRPTWMD is developed, and the problem is solved using Tabu Search and Ant Colony Optimization. The mathematical model described in this work is the main basis of the model described in Section 2. In Pureza et al. [2012], the fleet is homogeneous (all vehicles have the same characteristics, such as load capacities, travelling speed, and maximum crew size), and with either pickup or delivery for an entire route, exclusively, with a maximum number of available employees for the entire fleet. The authors also explore classic constraints, such as time windows, maximum cargo load, and maximum operational time. The objective function described aims to minimize the number of vehicles, the number of deliverymen, and the total distance traveled, in that order of importance. An example of a possible solution for the VRPTWMD is presented in Figure 1. In Figure 1, a vehicle leaves the depot (triangle) and parks at certain parking sites (red dots). Some of those parking sites have clients (black dots) nearby, *clusters* of clients (dotted areas). The deliverymen serve the clients at that cluster, return to the parking site, and move on to the next parking site in the route.

Figure 1: Possible solution for the VRPTWMD



Source: Pureza et al. [2012]

Several other works have studied this problem as well. Alvarez and Munari [2017] combine Iterated Local Search and Large Neighborhood Search with an exact branch-price-and-cut method to create a hybrid method to solve the VRPTWMD. In Souza Neto and Pureza [2016], the authors model the MTVRPTWMD (Multiple Trip VRPTWMD), a variation of the VRPTWMD where heterogeneous vehicles can be reused. Using a GRASP (Greedy Randomized Adaptive Search Procedure) heuristic with the solver CPLEX, the authors use data from a real company and generate solutions that reduced costs up to 37%. The works thus far assume that the client clusters already exist. Identifying clusters was studied in De Grancy [2015] and De Grancy and Reimann [2015], which both create the clusters and determine the vehicles' routes for the VRPTWMD.

Simultaneous Pickup and Delivery, described in Min [1989], allows clients to have both a delivery and a pickup demands. In this problem, each truck must leave the depot with all the cargo it will deliver, and must return with all the cargo it has picked up. This type of routing problem can be very useful in reverse logistics [Dethloff, 2001], where products can be delivered and picked up again for recycling purposes. Reverse logistics refers to the flow of products from the consumer back to the producer [Rogers and Tibben-Lembke, 2001]. This can be applied to recycling problems, garbage collection, repairs, re-manufacturing (for instance, product recall), and others. In the context of delivering beverages, the recycling problem fits perfectly, since there is not only a demand to deliver cases of drinks, but also of picking up empty bottles which are taken back to the factory for recycling.

The VRP with Simultaneous Pickup and Delivery is a well-established problem in the literature, and many works have already studied different approaches to solving it, such as: Local Search [Subramanian et al., 2010]; Particle Swarm [Kachitvichyanukul et al., 2009]; savings heuristics [Çatay, 2010]; Ant Colony Optimization [Gajpal and Abad, 2009]. Berbeglia et al. [2007] report on the general structure of the problems and detail a classification system for them.

This work explores the Heterogeneous Vehicle Routing Problem with Multiple Deliverymen and Simultaneous Pickup and Delivery (HVRPMDSPD). The HVRPMDSPD is a variation of the VRPTWMD where, aside from looking at time windows and multiple deliverymen, the fleet of vehicles is heterogeneous and each client can have both pick up and delivery demands. The

HVRPMDSPD was modeled as a Mixed Integer Linear Programming problem, and a constructive heuristic was developed to generate better initial solutions for the solver CPLEX. The model presented in this article is based on two main sources. The first one is the model in Pureza et al. [2012], which presents the VRP with Time Windows and Multiple Deliverymen. The second source is the work of Kachitvichyanukul et al. [2009], which describes a mathematical model for the VRP with Simultaneous Pickup and Delivery.

This work is divided as follows: Section 2 presents the MILP model developed; Section 3 describes the constructive heuristic used to generate better initial solutions for the solver CPLEX; Section 4 describes the computational experiments performed and shows their results; and finally Section 5 presents some concluding remarks and future works.

2. The mathematical model for the HVRPMDSPD

This section describes the Mixed Integer Linear Programming model developed for this work. This model aims to determine the routing strategy that serves all clients while minimizing the number of vehicles used, the number of deliverymen in the route, and the total distance traveled by the vehicles. The parameters of this model, its variables, constraints, and objective function are described as follows.

Parameters

N: Number of clients plus the depot, with index 1 being the depot;
K: Number of vehicles available in the fleet;
L^k: Maximum number of deliverymen in a given vehicle k;
M: Number of deliverymen available for the company;
q_i: Delivery demand of client i;
p_i: Pickup demand of client i;
$[a_i, b_i]$: Time window to start the service at client i;
tv_{ij}: Travel time between clients i and j;
ts_{il} Service time with l deliverymen at client i;
Q^k: Maximum cargo load of a vehicle k;
B_{ijl}: A large enough number;
c_1^k: Cost of using a vehicle k;
c_2: Cost of employing a deliveryman in a vehicle;
c_3: Cost associated with the distance traveled by each vehicle.

Moreover, $tv_{ij} = \frac{d_{ij}}{vel}$, where d_{ij} is the euclidean distance between i and j and vel is the vehicles' average speed, which is 1, as a convention. B_{ijl} is defined as in Pureza et al. [2012]:

$$B_{ijl} = max(b_i + ts_{il} + tv_{ij} - a_j, 0) \tag{1}$$

ts_{il} is also defined with a modification of the equation in Pureza et al. [2012], which guarantees that the service time is directly proportional to the cargo being delivered and picked up, and inversely proportional to the number of deliverymen, where rs is the rate of service of a deliveryman, which is 2 by convention:

$$ts_{il} = \frac{1}{l} min((q_i + p_i) * rs, b_1 - max(a_i, tv_{1i}) - tv_{i1}) \tag{2}$$

Variables

x_{ijlk}: $\begin{cases} 1, \text{ if vehicle } k \text{ leaves client } i \text{ and goes to } j \text{ with } l \text{ deliverymen;} \\ 0, \text{ otherwise.} \end{cases}$

y_{ilk}: total cargo in vehicle k, with l deliverymen, as it leaves client i.

t_{il}: instant in which service in client i begins with l deliverymen.

Flow constraints

Constraints (3), (4), and (5) guarantee the flow of vehicles between each client, while also ensuring that every client is served. Constraints (6) state that not every vehicle has to be used in the solution.

$$\sum_{\substack{i=1 \\ i \neq j}}^{N} \sum_{l=1}^{L^k} \sum_{k=1}^{K} x_{ijlk} = 1, \quad 2 \leq j \leq N; \tag{3}$$

$$\sum_{\substack{j=1 \\ j \neq i}}^{N} \sum_{l=1}^{L^k} \sum_{k=1}^{K} x_{ijlk} = 1, \quad 2 \leq i \leq N; \tag{4}$$

$$\sum_{\substack{j=1 \\ j \neq i}}^{N} x_{ijlk} = \sum_{\substack{j=1 \\ j \neq i}}^{N} x_{jilk}, \quad 1 \leq i \leq N, \quad 1 \leq l \leq L^k, \quad 1 \leq k \leq K; \tag{5}$$

$$\sum_{j=2}^{N} \sum_{l=1}^{L^k} x_{1jlk} \leq 1, \quad 1 \leq k \leq K; \tag{6}$$

Time constraints

Constraints (7) establish the beginning of the service in client j if a vehicle k with l deliverymen travels from i to j, taking into account the travel time between i and j and the service time in i with l deliverymen.

$$t_{jl} \geq t_{il} + (ts_{il} + tv_{ij}) x_{ijlk} - B_{ijl} * (1 - x_{ijlk}),$$
$$2 \leq i \leq N, \quad 1 \leq j \leq N, \quad i \neq j, \quad 1 \leq l \leq L^k, \quad 1 \leq k \leq K; \tag{7}$$

Load constraints

Constraints (8) and (9) update the cargo in vehicle k after leaving client j, coming from client i, with l deliverymen, calculating the current cargo, taking into account the pickup and delivery demands of client j. Constraints (10) state that the vehicle k must leave the depot with all the cargo to be delivered during its route.

$$y_{jlk} \geq y_{ilk} + (p_j - q_j) x_{ijlk} - Q^k (1 - x_{ijlk}),$$
$$1 \leq i \leq N, \quad 2 \leq j \leq N, \quad i \neq j, \quad 1 \leq l \leq L^k, \quad 1 \leq k \leq K; \tag{8}$$

$$y_{jlk} \leq y_{ilk} + (p_j - q_j)x_{ijlk} + Q^k(1 - x_{ijlk}),$$
$$1 \leq i \leq N, \quad 2 \leq j \leq N, \quad i \neq j, \quad 1 \leq l \leq L^k, \quad 1 \leq k \leq K; \quad (9)$$

$$y_{1lk} = \sum_{i=1}^{N} \sum_{\substack{j=1 \\ i \neq j}}^{N} q_i x_{ijlk}, \quad 1 \leq l \leq L^k, \quad 1 \leq k \leq K; \quad (10)$$

Deliverymen constraint
Constraint (11) ensures that the total number of deliverymen to leave the depot does not exceed the number of deliverymen available.

$$\sum_{j=2}^{N} \sum_{l=1}^{L^k} \sum_{k=1}^{K} l x_{1jlk} \leq M \quad (11)$$

Variable Domains
Constraints (12), (13), (14), (15), define the domains of the variables in the model. Constraints (16) also limits the beginning of service at each client i by the time window of that client.

$$x_{ijlk} \in \{0, 1\}, \quad 1 \leq i \leq N, \quad 1 \leq j \leq N, \quad i \neq j, \quad 1 \leq l \leq L, \quad 1 \leq k \leq K; \quad (12)$$
$$y_{ilk} \in Z^+, \quad 1 \leq i \leq N, \quad 1 \leq l \leq L, \quad 1 \leq k \leq K; \quad (13)$$
$$max\{p_i - q_i, 0\} \leq y_{ilk} \leq Q^k, \quad 1 \leq i \leq N, \quad 1 \leq l \leq L, \quad 1 \leq k \leq K; \quad (14)$$
$$t_{il} \in R^+, \quad 1 \leq i \leq N, \quad 1 \leq l \leq L; \quad (15)$$
$$a_i \leq t_{il} \leq b_i, \quad 1 \leq i \leq N, \quad 1 \leq l \leq L; \quad (16)$$

Thus, the MILP model for the HVRPMDSPD can be written as follows:

$$\min Z = c_1^k \sum_{j=2}^{N} \sum_{l=1}^{L^k} \sum_{k=1}^{K} x_{1jlk} + c_2 \sum_{j=2}^{N} \sum_{l=1}^{L^k} \sum_{k=1}^{K} l x_{1jlk} + c_3 \sum_{i=1}^{N} \sum_{\substack{j=1 \\ i \neq j}}^{N} \sum_{l=1}^{L^k} \sum_{k=1}^{K} d_{ij} x_{ijlk} \quad (17)$$

subject to: (3) ... (16)

The objective function (17) aims to minimize the total operational cost, where c_1^k is the cost of using a vehicle k, c_2 is the cost of employing a deliveryman, and c_3 is the cost associated with the distance traveled.

3. Constructive heuristic

In order to generate initial solutions, a constructive algorithm based on the method used in Alvarez and Munari [2016] was developed. This method constructs each route by greedily adding to a given route the client that is furthest away from the depot, maintaining feasibility. The pseudo-code for this constructive method is shown in Algorithm 1. The results obtained using this strategy will be discussed in Section 4.

Algorithm 1 Constructive algorithm

```
1:  procedure CONSTRUCTROUTES
2:      DistList ⇐ OrderClientsFurthestAway()
3:      Vis ⇐ 1                                              ▷ Initializes Vis with the depot
4:      for v from 1 to NumV do                              ▷ For every vehicle in the fleet
5:          route[v][0] ⇐ 1          ▷ The first value refers to the number of deliverymen in that route
6:          for each NewCl in DistList do
7:              if NewCl is not in Vis then                  ▷ If client NewCl hasn't been visited yet
8:                  FinalPos ⇐ null
9:                  NumDel ⇐ L[v] + 1                        ▷ Maximum number of deliverymen in vehicle v
10:                 Savings ⇐ MaxInt
11:                 for j from 0 to length(route[v]) do
12:                     fact, del ⇐ VerifyFeasibility(NewCl,j,route[v])      ▷ del - deliverymen required
13:                     if fact = True then
14:                         pos ⇐ j
15:                         prev ⇐ 1
16:                         if pos ≠ 0 then
17:                             prev ⇐ route[v][pos]
18:                         next ⇐ 1
19:                         if pos ≠ len(route[v] -1) then
20:                             next ⇐ route[v][pos + 1]
21:                         sv ⇐ dist[prev][NewCl] + dist[NewCl][next] - dist[prev][next]
22:                         if sv < Savings or del < NumDel then
23:                             Savings ⇐ sv
24:                             FinalPos ⇐ pos
25:                             NumDel ⇐ del
26:                 if FinalPos ≠ null then                  ▷ Client i can successfully be added to the route
27:                     route[v] ⇐ NewRoute (route[v],NumDel, FinalPos, NewCl)
28:                     Vis ⇐ Vis ∪ NewCl
29:     return route
```

Algorithm 1 details the steps taken in order to construct a solution for the HVRPMDSPD. In Line 2, a function called *OrderClientsFurthestAway()* returns a list of clients sorted by their distance to the depot. Thus, the first client in *DistList* is the one furthest away from the depot. This ensures the order in which the clients are inserted in the routes. The *for* loop in Line 4 looks at every individual vehicle and constructs their routes. It is important to note that the vehicles are ordered from least costly, to most costly. This means that the constructive heuristic tries to fill up the cheaper vehicles first, and only later looks at the more expensive ones. Line 5 ensures that the first element in *route[v]* refers to the number of deliverymen in that route. This information is crucial for the feasibility verification in Line 12. The *for* loop in Line 11 is called whenever a client has not been assigned to a route yet. This loop attempts to find the best position in the route to insert said client, if that position exists. Line 12 calls a function which verifies whether the new client *NewCl* can be inserted in position *j*, and, if so, how many deliverymen are necessary for that insertion to be feasible. Line 21 calculates the cost in total distance traveled of inserting client *NewCl* in position *j*. This approach is inspired by the work of Clarke and Wright [1964]. Line 22 decides whether this insertion is the new best position possible to insert client *NewCl*. The criterion is that this new insertion position should either provide better savings or use fewer deliverymen. This step ensures that, by the end of the *for* loop, the method will have found the best position to insert *NewCl*, if that

position exists. Finally, if an insertion position has been found, Line 27 updates the route, and Line 28 updates the list of clients that have already been visited, so they may not be considered again in other routes.

4. Computational experiments

In order to validate the model and test the constructive heuristics, instances with 25, 50, and 100 clients were solved using the solver CPLEX 12.8 with a Python program implementing the model described in Section 2.

The problems were solved using a machine with 2 processors Intel Xeon E5-2680v2 with 2.8 GHz 10 cores and 128 GB DDR3 1866MHz of memory, with a time limit of 30 minutes. The instances solved were the *rdp101*, *rdp102*, *rdp103*, *rdp104*, and *rdp105* instances with 100 clients, and the *rcdp25*, *rcdp50*, with 25 and 50 clients each [Wang and Chen, 2012]. These instances are an adaptation of instances in Solomon [1987], with simultaneous pickup and delivery and time windows. In order to calculate the service time at each client, an extension of the calculation in Pureza et al. [2012] was used, as shown in Equation 2. The parameters used were based on parameters established in the literature, $M = N - 1$, $vel = 1$, $rs = 2$, e $K = N - 1$, where N is the number of clients plus the depot.

The costs associated with using a vehicle vary with the type of vehicle, and are shown in Tables 1 and 2. In Pureza et al. [2012], $c_1 = 1$, with a homogeneous fleet, $c_2 = 0.1$, the cost of using a deliveryman, and $c_3 = 0.0001$, the cost of the distance traveled. In this article, since the fleet is heterogeneous, $c_2 = 0.1 * c_1^0$, or 0.1 times the cost of the cheapest vehicle, and $c_3 = 0.0001 * c_1^0$, or 0.0001 times the cost of the cheapest vehicle. Moreover, in order to minimize the number of variables and constraints, a simple pre-processing was used. In this pre-processing, the variables x_{ijlk} describing an arc where $a_i + ts_{il} + tv_{ij} > b_j$ were not created, because those arcs are not feasible and break the time windows.

Additionally, the types of vehicles were defined as in Liu and Shen [1999]. In that work, for every set of Solomon instances [Solomon, 1987] R1 and R2, C1 and C2, RC1 and RC2, different vehicle types were defined, with varying cargo and costs. For every set of instances, the authors also determined three costs variations. These variations consist of one with high cost (type *a*), medium cost (type *b*), and low cost (type *c*). Tables 1 and 2 present the vehicle and cost types for the R1 and RC1 instances (studied in this article), varying on a maximum number of deliverymen, maximum load, and alternative costs.

Table 1: Vehicle types for instances R1

			Costs		
Type	L	Load	R1a	R1b	R1c
A	1	30	50	10	5
B	2	50	80	16	8
C	2	80	140	28	14
D	3	120	250	50	25
E	3	200	500	100	50

Table 2: Vehicle types for instances RC1

			Costs		
Type	L	Load	RC1a	RC1b	RC1c
A	1	40	60	12	6
B	2	80	150	30	15
C	2	150	300	60	30
D	3	200	450	90	45

Table 3 shows the results obtained without the initial solutions provided by the constructive heuristic within the time limit, with the values of the objective function, the number of vehicles and deliverymen and total distance traveled, as well as total execution time and the gap. The results

Table 3: Results found by the solver CPLEX without the constructive heuristic

Instance	Class	Objective Function	Nº Vehicles	Nº Deliverymen	Total Distance	Time (s)	GAP
rcdp2501	a	1115.57	11	15	928.88	TL	21.89
rcdp2501	b	229.03	10	15	857.11	TL	23.90
rcdp2501	c	127.77	12	17	949.60	TL	31.80
rcdp2504	a	2411.59	9	16	932.47	TL	99.93
rcdp2504	b	374.26	8	16	884.44	TL	99.91
rcdp2504	c	201.43	8	15	716.55	TL	99.91
rcdp2507	a	2136.20	12	20	1033.55	TL	96.78
rcdp2507	b	473.83	9	19	859.70	TL	94.32
rcdp2507	c	234.58	10	20	971.26	TL	97.10
rcdp5001	a	2632.56	22	32	1759.54	TL	36.40
rcdp5001	b	452.13	21	30	1775.88	TL	26.58
rcdp5001	c	232.72	22	31	1859.02	TL	30.12

show that without the constructive heuristic, the solver was unable to find any solutions for instances *rcdp5004* and *rcdp5007*, with 50 clients. It was also unable to find solutions for any instance with 100 clients in the time limit established. Table 4 presents the results for the same instances using the constructive heuristic. All executions ran until the time limit of 30 minutes.

Table 4: Results found by the solver CPLEX using the constructive heuristic

Instance	Class	Objective Function	Nº Vehicles	Nº Deliverymen	Total Distance	Time (s)	GAP
rcdp2501	a	1109.55	11	14	924.65	TL	18.14
rcdp2501	b	223.15	11	15	959.83	TL	21.90
rcdp2501	c	121.17	11	16	944.32	TL	28.08
rcdp2504	a	1110.44	11	14	1074.00	TL	99.84
rcdp2504	b	222.08	11	14	1068.21	TL	99.84
rcdp2404	c	111.05	11	14	1085.90	TL	99.84
rcdp2507	a	1441.40	13	19	1233.26	TL	95.27
rcdp2507	b	255.78	12	17	1150.11	TL	94.65
rcdp2507	c	137.50	12	18	1161.41	TL	90.21
rcdp5001	a	2490.09	23	33	2015.78	TL	34.70
rcdp5001	b	496.86	23	32	2046.93	TL	34.54
rcdp5001	c	235.71	21	31	1850.81	TL	30.52

Some solutions found by the solver CPLEX without a constructive heuristic seem better than the ones found with the heuristic. For instance *rcdp2507b*, the solver by itself found a solution with 9 vehicles and 19 deliverymen, whereas the solution found with the heuristic start used 12 vehicles and 17 deliverymen. However, analyzing strictly the objective function, it is clear that the constructive heuristic helps the solver find better solutions overall, considering the trade-off between using more vehicles and employing less deliverymen. In fact, the results using the constructive heuristic are only worse in two instances, *rcdp5001b*, and *rcdp5001c*, and, even then, the results are

only worse at most by a margin of 10%, as can be seen in Tables 6 and 7. Conversely, the results obtained without using the constructive heuristic consistently provide a worse trade-off between the number of vehicles and the number of deliverymen.

Moreover, the fact that, without the constructive heuristic, the solver could not find solutions for most of the instances before the time limit shows that the heuristic is useful, especially for large-scale instances. Table 5 shows the results obtained for the instances for which the solver CPLEX without the constructive heuristic could not find any solutions within the time limit.

Table 5: Instances for which the solver could not find solutions without the heuristic within the time limit

Instance	Class	Objective Function	Nº Vehicles	Nº Deliverymen	Total Distance	Time (s)	GAP
rcdp5004	a	2166.82	21	29	2136.38	TL	99.91
rcdp5004	b	433.39	21	29	2155.27	TL	99.91
rcdp5004	c	216.69	21	29	2145.77	TL	99.91
rcdp5007	a	2160.92	21	28	2153.88	TL	99.89
rcdp5007	b	432.18	21	28	2153.88	TL	99.89
rcdp5007	c	216.09	21	28	2156.24	TL	99.89
rdp101	a	3797.12	46	68	3424.48	TL	32.92
rdp101	b	750.33	45	69	3332.27	TL	32.10
rdp101	c	379.71	46	68	3424.48	TL	100.00
rdp102	a	3503.00	44	65	3599.14	TL	100.00
rdp102	b	700.60	44	65	3599.14	TL	100.00
rdp102	c	350.30	44	65	3599.14	TL	100.00
rdp103	a	3187.77	42	58	3554.44	TL	98.17
rdp103	b	637.55	42	58	3554.44	TL	98.17
rdp103	c	318.78	42	58	3554.44	TL	98.17
rdp104	a	3193.08	42	59	3616.17	TL	98.18
rdp104	b	638.62	42	59	3616.17	TL	98.18
rdp104	c	319.31	42	59	3616.17	TL	98.18
rdp105	a	3343.11	43	61	3622.60	TL	100.00
rdp105	b	668.62	43	61	3622.60	TL	100.00
rdp105	c	334.31	43	61	3622.60	TL	100.00

Table 6: Comparison between the objective values obtained with and without the constructive heuristic

	rcdp2501a	rcdp2501b	rcdp2501c	rcdp2504a	rcdp2504b	rcdp2504c	rcdp2507a	rcdp2507b	rcdp2507c
Cplex	1115.57	229.03	127.77	2411.59	374.26	201.43	2136.20	473.83	234.58
Cplex and heuristic	**1109.55**	**223.15**	**121.17**	**1110.44**	**222.08**	**111.05**	**1441.40**	**255.78**	**137.50**
Comparison	-0.54%	-2.57%	-5.17%	-53.95%	-40.66%	-44.87%	-32.53%	-46.02%	-41.39%

Table 7: Comparison between the objective values obtained with and without the constructive heuristic

	rcdp5001a	rcdp5001b	rcdp5001c
Cplex	2632.56	**452.13**	**232.72**
Cplex and heuristic	**2490.09**	496.86	235.71
Comparison	-5.41%	+9.89%	+1.29%

5. Concluding remarks

This article aimed to describe, define, model, and provide solutions for Heterogeneous Vehicle Routing Problem with Multiple Deliverymen and Simultaneous Pickup and Delivery (HVRPMDSPD). In this problem, a company must pick up and deliver goods at clients' locations during specific time windows. Some clients can also be close to another, thus, instead of parking at each client's location, the vehicle stops at a parking site nearby, from where the deliverymen inside perform the service on foot, to cut down on service times.

A Mixed Integer Linear Programming model for the HVRPMDSPD was developed, aiming to minimize operational costs of using vehicles, employing deliverymen, and the total distance travelled. Instances for the VRP with Simultaneous Pickup and Delivery with Time Windows from Wang and Chen [2012] were combined with Heterogeneous VRP instances from Liu and Shen [1999] and solved by the solver CPLEX. The HVRPMDSPD has never been discussed or solved in the literature, and thus there are no results available for comparison and validation. In order to provide better starting solutions, a constructive heuristic based on the work of Alvarez and Munari [2016] was developed and its solutions were given to the solver CPLEX. A comparison between the results with and without using the constructive heuristic was performed so as to validate this heuristic. Results showed that using the constructive heuristic the solver CPLEX could find better solutions than without using it. Additionally, the constructive heuristic allowed the solver to determine solutions for other instances with 50 clients and instances with 100 clients for which no solution was found before the time limit without the heuristic.

In the future, a more complete meta-heuristic, such as the Iterated Local Search, could be explored, rather than just a constructive heuristic. Other characteristics for the problem can be investigated, such as maximizing the number of clients served, considering multiple trips, priorities between clients, or even multiple depots.

Acknowledgements

The authors would like to thank CNPq, CAPES (PROEX-10545229/M), and CeMEAI-CEPID (FAPESP No. 2013/07375-0) for the financial support received.

References

Alvarez, 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.

Alvarez, D. A. and Munari, P. (2016). Abordagens metaheurísticas para o problema de roteamento de veículos com janelas de tempo e múltiplos entregadores. *Gestão e Produção*, 23:279–293.

Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I., and Laporte, G. (2007). Static pickup and delivery problems: a classification scheme and survey. *Top*, 15(1):1–31.

Braekers, K., Ramaekers, K., and Van Nieuwenhuyse, I. (2016). The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering*, 99:300–313.

Çatay, B. (2010). A new saving-based ant algorithm for the vehicle routing problem with simultaneous pickup and delivery. *Expert Systems with Applications*, 37(10):6809–6817.

Clarke, G. and Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, 12(4):568–581.

De Grancy, G. S. (2015). An adaptive metaheuristic for vehicle routing problems with time windows and multiple service workers. *J. UCS*, 21(9):1143–1167.

De Grancy, G. S. 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.

Dethloff, J. (2001). Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up. *OR-Spektrum*, 23(1):79–96.

Gajpal, Y. and Abad, P. (2009). An ant colony system (acs) for vehicle routing problem with simultaneous delivery and pickup. *Computers & Operations Research*, 36(12):3215–3223.

Kachitvichyanukul, V. et al. (2009). A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, 36(5):1693–1702.

Liu, F.-H. and Shen, S.-Y. (1999). The fleet size and mix vehicle routing problem with time windows. *Journal of the Operational Research society*, 50(7):721–732.

Min, H. (1989). The multiple vehicle routing problem with simultaneous delivery and pick-up points. *Transportation Research Part A: General*, 23(5):377–386.

Pureza, V. and Morabito, R. (2010). Designando entregadores extras no roteamento de veículos com janelas de tempo. *Anais do XLII Simpósio Brasileiro de Pesquisa Operacional*.

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.

Rogers, D. S. and Tibben-Lembke, R. (2001). An examination of reverse logistics practices. *Journal of business logistics*, 22(2):129–148.

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. d. 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.

Subramanian, A., Drummond, L. M. d. A., Bentes, C., Ochi, L. S., and Farias, R. (2010). A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, 37(11):1899–1911.

Wang, H.-F. and Chen, Y.-Y. (2012). A genetic algorithm for the simultaneous delivery and pickup problems with time window. *Computers & Industrial Engineering*, 62(1):84–95.