INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

IFORS

Check for updates

WILEY

INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

Intl. Trans. in Op. Res. 30 (2023) 1031–1064
DOI: 10.1111/itor.13071

A new two-phase heuristic for a problem of food distribution with compartmentalized trucks and trailers

Laura Davila-Pena, David R. Penas and Balbina Casas-Méndez*

Group of Optimization Models, Decision, Statistics and Applications (MODESTYA), Department of Statistics, Mathematical Analysis and Optimization, Institute of Mathematics (IMAT), Faculty of Mathematics, Universidade de Santiago de Compostela, Santiago de Compostela, Spain
E-mail: lauradavila.pena@usc.es [Davila-Pena]; david.rodriguez.penas@usc.es [R. Penas]; balbina.casas.mendez@usc.es [Casas-Méndez]

Received 30 July 2020; received in revised form 13 September 2021; accepted 27 September 2021

Abstract

This paper presents a new formulation for the routing problem in which the available fleet consists of trucks and trailers divided into compartments. Solving the model for large instances is computationally expensive. Therefore, we introduce and implemented a two-phase heuristic algorithm. In the first phase, an initial solution is generated through a constructive heuristic algorithm based on concepts from the classic Clarke–Wright algorithm. In the second phase, the initial solution is improved by an iterated tabu search metaheuristic. Our algorithm was tested on 21 instances that were converted from the classic truck and trailer routing problem. The results of our computational study prove the effectiveness of our proposal; the algorithm always finds a feasible solution, which in small-sized problems it is proven to be of good quality. In addition, the algorithm outperforms previous approaches for some truck and trailer routing problem instances. Furthermore, an application of the proposed model and heuristic is demonstrated in the field of agricultural logistics by comparing the obtained results.

Keywords: truck and trailer routing problem; compartmentalized vehicles; construction heuristic algorithm; tabu search; logistics

1. Introduction

In recent years, transport logistics has played a fundamental role in industry. Many public and private companies are interested in developing computational tools to design their routes, with objectives such as minimizing costs and/or maximizing the distribution of products. Thus, vehicle routing problems (VRPs) are a popular type of combinatorial optimization problem, through which transport routes for vehicles visiting a set of customers located at different places can be modeled.

*Corresponding author.

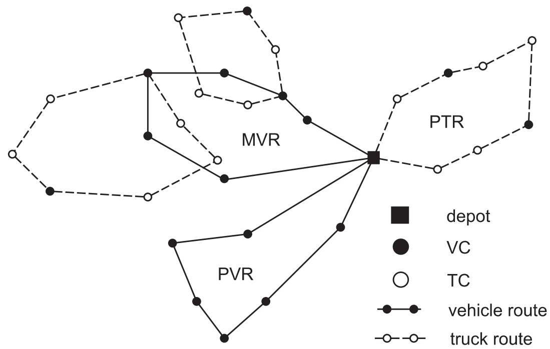


Fig. 1. Possible solution to the TTRP.

Solving these mathematical optimization problems is a challenge for operational researchers. When new features from real-world applications are considered, such as capacitated vehicles, delivering in limited time windows, and stochastic behaviors, new variations of the original VRP arise, creating a need to develop new models and solution techniques.

One promising modification of the VRP is the truck and trailer routing problem (TTRP) proposed by Chao (2002), which incorporates accessibility restrictions. In this variant, a fleet of trucks and trailers visits a set of customers, where some customers (vehicle customers; VCs) can be served by a complete vehicle (i.e., a truck pulling a trailer), while others are only reachable by a truck alone (truck customers; TCs). Examples of TCs are customers in inner-city areas, mountainous regions, or places where maneuvering or access with a trailer is not possible. To solve this problem, we distinguish three types of routes: pure truck routes (PTRs), which can be traveled only by trucks, pure vehicle routes (PVRs), which can be traveled entirely by a complete vehicle, and mixed vehicle routes (MVRs), which consist of a main tour traveled by a complete vehicle and one or more sub-tours traveled only by the truck part of the vehicle. Figure 1 illustrates a possible solution to the TTRP. Although this model can be very useful in many land-based logistical applications, the presence of three different types of routes makes solving the associated optimization problem more difficult, suiting it to the application of heuristics and metaheuristics, such as in Lin et al. (2009). Real-world applications include farm milk collection (Caramia and Guerriero, 2010b), delivery by feed mills (Lin et al., 2009), and the provisioning of infrastructure services in urban areas with accessibility restrictions (Parragh and Cordeau, 2017).

Another interesting variation of the VRP is the multi-compartment case, where different products must be split into different storages during transport, making it challenging to maximize the use of vehicle capacity on the generated routes. Although the inclusion of compartments adds extra complexity, it can be a requirement in real logistical applications, as explained by Guitián de Frutos and Casas-Méndez (2019). Therefore, this paper proposes a novel mixed integer linear programming (MILP) approach to combine the TTRP with product compartmentalization, which we call the multi-compartment TTRP (MC-TTRP). The combination of these two features is motivated by the needs of a Spanish agricultural cooperative that produces feed for cattle, which we used to test, illustrate, and apply the proposed formulation. A tentative MC-TTRP was originally introduced in an unpublished preliminary work by Davila-Pena (2019).

Meanwhile, the effectiveness of the Clarke–Wright algorithm (Clarke and Wright, 1964) in building a solution for different VRPs and the requirement to solve MC-TTRPs that involve a relatively high number of customers served as motivation to modify this heuristic algorithm for the case of the MC-TTRP. To the best of our knowledge, only Derigs et al. (2013) reported adaptation of this algorithm to build an initial TTRP solution, although no details about such adaptation were provided. In addition to this constructive method, we propose a metaheuristic approach based on an iterated tabu search to improve the initial solution obtained. Both the constructive and improvement phases are integrated into a novel two-stage algorithm to solve the MC-TTRP. A corresponding computational study was conducted through a series of instances created from other existing ones in the literature, obtaining excellent results. However, these problems could not be benchmarked with the exact model due to the computational time required. In contrast, a series of small-sized real-world problems were solved, achieving solutions that are competitive and close to those provided by the exact method.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 presents the current case study in detail and an in-depth description and formulation of the MC-TTRP. Section 4 presents the two-stage heuristic to solve the MC-TTRP. Section 5 reports the computational results for the designed heuristic on the MC-TTRP, using a set of instances adapted from those in literature and data of a real application. Finally, Section 6 summarizes the main conclusions of our study.

2. Related work

To solve the logistics of an agricultural cooperative that distributes feed for cattle to a large number of customers, many of whom have accessibility restrictions, the TTRP appears to be a satisfactory model. Although Chao (2002) introduced the term TTRP, previous works have incorporated trailers to solve similar case studies. The first approach could be that presented by Semet and Taillard (1993). These authors proposed a VRP that considered the use of trailers under accessibility restrictions. Semet (1995) proposed another example describing a new variant of the VRP formulated as an integer linear programming (ILP) problem called the partial accessibility constrained VRP (PACVRP). Despite being very similar to Chao's TTRP, it has specific differences, such as the utilization of all available trucks. Other studies have considered a heterogeneous fleet of vehicles composed of trucks and trailers, such as the case of Gerdessen (1996), whose model is known as the VRP with trailers (VRPT). Moreover, Chao et al. (1998) studied the site-dependent VRP (SDVRP), where every customer has a specific type of vehicle assigned. Some seminal papers on the TTRP do not offer a mathematical formulation through an MILP model, although Scheuerer (2004) presented a formulation of the TTRP by Chao (2002). This turns out to be an adaptation of the proposal by Semet (1995) for the PACVRP and can be considered as the motivation for the current paper.

Other researchers have built new models based on the proposal of Chao (2002) to meet various real-world requirements, such as a TTRP with time windows (TTRPTW) proposed by Lin et al. (2011). In the TTRPTW, besides its type and demand, each customer has three associated measurable times: the earliest and latest time of day at which it can be served and the service time required. Recently, Accorsi and Vigo (2020) considered a generalization of the TTRP, the extended single

TTRP (XSTTRP), which contains, all together, a variety of node types that were previously considered only separately: truck customers, vehicle customers with and without parking places, and parking-only locations. In the XSTTRP, a single vehicle, consisting of a truck and a detachable trailer, is used to serve a set of customers with known demand and accessibility constraints.

Regarding TTRP solution methods, heuristics are popular approximation-based strategies for solving medium- to large-scale instances. In fact, heuristics have been used in the solutions of several VRP variants, with Lespay and Suchan (2021) being one of the most recent references. That study considered the problem of a food company's distribution center. This was solved by constructing an initial solution, which was subsequently improved using a guided local search. Gerdessen (1996) proposed constructive and improvement heuristics for solving the VRPT. Semet (1995) described a two-stage heuristic method for obtaining PACVRP solutions: the first phase of the algorithm involves assigning trailers to trucks and determining the optimal allocation between customers and trucks/vehicles, and then, the second phase builds the routes. Other works, such as those by Chao (2002) and Scheuerer (2006), also proposed two-phase methods, where they first defined an initial solution by applying constructive procedures and then used improvement metaheuristics based on techniques such as tabu search (Glover and Laguna, 1998). Later, Caramia and Guerriero (2010a) combined a mathematical programming and local search approach to solve the TTRP, and they compared their results with those of Chao (2002) using a set of benchmarks. Furthermore, the TTRP can be addressed using a metaheuristic approach, as in the work by Lin et al. (2011), where a simulated annealing algorithm was designed to find approximate TTRP solutions according to given time windows, achieving improved results in 11 of the 21 instances of Chao (2002). In addition, in an original research, Derigs et al. (2013) analyzed different variants of the TTRP and proposed two-stage heuristics for solving these problems, starting by building an initial solution and then moving to an improvement phase combining techniques such as local search (LS) and large neighborhood search (LNS). The behavior of the heuristics created for the TTRPTW were compared with the heuristic proposed by Lin et al. (2011). Depending on the TTRP variant considered, the authors applied a specific construction heuristic and, among them, an adaptation of the Clarke–Wright savings algorithm stood out. In terms of exact solution methods, recent references include the paper by Parragh and Cordeau (2017), which proposes a branching and pricing algorithm for the TTRPTW. It adapts the LNS algorithm to obtain good initial columns. Compared with existing metaheuristic algorithms, such as those designed by Lin et al. (2011) and Derigs et al. (2013), they obtained highly competitive results. Some instances with up to 100 customers were optimally solved. Rothenbächer et al. (2018) also solved the TTRPTW exactly using a branching, bounding, and cutting algorithm. Their computational studies showed that their algorithm outperforms existing approaches on the TTRP and TTRPTW benchmark instances used in the literature. To solve the XSTTRP, Accorsi and Vigo (2020) developed a fast and efficient hybrid metaheuristic based on a four-phase solution approach, in which the main improvement phase consists of an iterated local search.

Another challenging variation of the VRP arises when customers demand various types of products that cannot be mixed. This is the case in the multi-compartment VRP (MC-VRP), which was initially presented in Brown and Graves (1981) and Brown et al. (1987), whose objective was the distribution of petroleum products in the United States.

Concerning the solving of multi-compartment problems, different approaches have been followed in recent years based on heuristics and metaheuristics. Simple constructive algorithms, such as the

Clarke–Wright algorithm, have also been successfully adapted in this context, as can be seen in the literature. El Fallahi et al. (2008) compared a constructive algorithm, memetic algorithm, and tabu search, concluding that the results provided by the tabu search were slightly better, although it required more computation time. Muyldermans and Pang (2010) used the Clarke–Wright savings algorithm to obtain a feasible initial solution. Subsequently, they performed a local search with movements taken from the literature and improved the quality of the solution previously obtained through a metaheuristic based on a guided local search. They performed a sensitivity analysis on certain parameters (number of customers and their demands, depot location, vehicle capacity, or number of products). Their computational study included a comparison with the work of El Fallahi et al. (2008). Derigs et al. (2011) considered a model with a homogeneous fleet, that is, all vehicles have the same number of compartments, all with equal capacities. This problem is a particular case of that addressed in the current paper. They implemented their own benchmarks and a collection of optimization methods capable of obtaining high-quality solutions, which covered a wide range of alternative approaches to construction, such as LS, LNS, and metaheuristics. Coelho and Laporte (2015) defined and compared four categories of multi-compartment problems. They proposed two formulations for each case and presented a branching and cutting algorithm to solve single- and multi-period cases containing up to 50 and 20 customers, respectively. Mendoza et al. (2010) extended the MC-VRP to the case in which the demands are stochastic, giving rise to the MC-VRP with stochastic demands (MC-VRPSD). Mendoza et al. (2011) proposed a set of constructive heuristics to solve this problem, which included stochastic versions of the nearest neighbor, nearest insertion, and savings-based approaches, adapted to the multi-compartment scenario.

Among the most recent investigations of solution methods for the MC-VRP, we highlight those by Henke et al. (2015, 2019). Starting from a real problem of collecting glass containers, a model formulation and branch-and-cut algorithm for solving the problem to optimality were presented. The performance of the proposed algorithm was evaluated through extensive numerical experiments. Furthermore, the economic benefits of introducing compartments to vehicles were investigated. Silvestrin and Ritt (2017) proposed a tabu search heuristic algorithm and integrated it with an iterated local search to solve the MC-VRP. In several experiments, they analyzed the performance of the algorithm and compared it with results in the literature, finding that it produces better solutions than those provided by other existing heuristic algorithms. They considered an initial solution obtained by the Clarke–Wright savings algorithm extended to handle multiple compartments. Metaheuristics based on iterated local searches have shown very good behavior in various VRP variants (cf. Alvarez et al., 2018, who proposed efficient metaheuristics based on iterated local search and simulated annealing). Alinaghian and Shokouhi (2018) presented a new mathematical model for the multi-depot MC-VRP. They designed a hybrid algorithm composed of adaptive large neighborhood search (ALNS) and variable neighborhood search (VNS). The results were compared to the exact solutions of small instances and compared with each other in large instances. Ostermeier and Hübner (2018) proposed an MC-VRP with a fleet of vehicles with flexible compartments. The aim of their work was to demonstrate the benefits of considering a mixed fleet consisting of both single-compartment and compartmentalized vehicles. The problem was solved using LNS. Ostermeier et al. (2021) introduced a typology for MC-VRPs and extensively reviewed the existing literature. They also made suggestions for future research.

Finally, the work of Caramia and Guerriero (2010b) should be highlighted as the first (and to the best of our knowledge, the only) to consider the TTRP with compartments, which we refer to as the MC-TTRP hereinafter. They investigated a VRP in which at most one type of product could be assigned to each compartment. Furthermore, they established the additional constraint that some delivery locations were small and inaccessible by large vehicles. Due to the similarity between the problem addressed in that paper and the present one, it is considered convenient to point out the differences between the two studies. First, with regard to actual motivation, the problem analyzed by Caramia and Guerriero (2010b) was for milk collection on farms by an Italian company, while in our real-world case study, which will be described in more detail in the next section, the problem of the distribution of feed among members of a Spanish agricultural cooperative was analyzed. Regarding the model and methodology used, Caramia and Guerriero (2010b) proposed two mathematical programming models. One of them aimed to assign vehicles to farmers with the objective of minimizing the number of vehicles used, satisfying restrictions on capacity, demand, and types of milk. It should be noted that the group of farmers was divided into four zones, and an initial allocation of vehicles was made to each zone. The fleet considered was heterogeneous. The second model was used to minimize the lengths of the routes. In the proposed methodology, the possibility of serving VCs on sub-tours was not permitted. Accordingly, they used a two-phase heuristic. Such a process might result in no feasible solutions with respect to the times of work shifts. Therefore, a multiple-restart mechanism was implemented, and additional constraints, local search, and a tabu list were added to avoid cycling. Following a different approach, in our setup, there is a homogeneous truck fleet and a trailer fleet, and vehicle pre-assignments are not made to groups of customers. In this case, the sub-tours on an MVR, traveled by only a truck, can visit both TCs and VCs. In addition, what makes an important difference is that the formulation of a single model covering the whole problem is provided. We solved the model exactly for small-sized instances and then developed a two-phase heuristic, in which a generalization of the Clarke–Wright algorithm is used to find an initial solution that is then improved by a tabu search. The results of the heuristic were compared to the optimal solutions in problems where it was possible to do so, and a comprehensive computational study was conducted by creating MC-TTRP test problems for the heuristic. It is also worth mentioning that we studied the performance of our heuristic on TTRP instances, in addition to studying the scope of our model with instances built from real data.

3. Problem description and formulation

3.1. Case study

The motivation for this study stemmed from the needs of a Spanish cooperative that produces and distributes feed for farm animals. The company is located in Galicia, a region in the northwestern Spain with an area of 29,565 km^2 spread over four provinces and 315 municipalities. The cooperative, which was created 16 years ago, currently has a total clientele of more than 1500 farmers distributed throughout the four provinces of Galicia (although not all of them order from the feed factory) and covering 60 municipalities across a large geographical area. The annual amount of feed produced exceeds 150,000 tons.

The agricultural company produces different types of feed, and farmers usually place one or two orders per month. The number of daily orders is approximately 40, where each order ranges from 500 to more than 14,000 kg. The average number of annual orders per feed customer is approximately 17. There are also occurrences such as the loss or incorporation of new customers. The roads leading to some of the farms or the farms themselves are inaccessible by large trailers. Moreover, customers sometimes request different types of feed because they have different species of animals. Naturally, goods that are not of the same type cannot be mixed. Thus, it is necessary to have compartmentalized vehicles. In addition to not being able to mix different kinds of feed in the same hopper, the same compartment cannot be used to supply two different customers because the cooperative does not have technology to measure out each customer's supply from their vehicles.

The purpose of this study was to provide a tool for the cooperative to automatically design routes for each vehicle such that their restrictions are met and the distance traveled is minimized. Each day, new orders may be received, trucks may experience breakdowns, and customers may change their demands at short notice. All these factors suggest that route planning is only useful within two or three days at most.

A team of agricultural engineers designed a comprehensive global positioning system (GPS) that can monitor various vehicle routes. The GPS provides all the geographic information required to provide the data to solve the problem. We also know the capacity of each compartment, the demands of different customers, and whether a trailer can access each farm as well as its load restrictions.

3.2. Multi-compartment truck and trailer routing problem (MC-TTRP)

As stated before, this paper proposes an MC-TTRP model—a novel MILP implementation of the TTRP with multi-compartmentalized vehicles.

The MC-TTRP can be described as follows. Let $G = (N, E)$ be an undirected, weighted graph consisting of a node set $N = \{0, 1, \dots, n\}$, representing the depot ($\{0\}$) and customers ($\{1, \dots, n\}$), and an arc set $E = \{(i, j) : i, j \in N, i \neq j\}$, representing the arcs that can be traveled between different nodes. N_1 and N_2 are subsets of N that contain the n_1 and $n - n_1$ VCs and TCs, respectively. A nonnegative cost c_{ij}, $(i, j) \in E$, is assigned to each arc, which represents the distance a vehicle must travel from i to j. Each node $i \in N$ requires a service time s_i, which, in the case of the depot, refers to the time required to load the vehicles. For transportation, a set $K^T = \{1, \dots, m_L, \dots, m_T\}$ of trucks and set $K^L = \{1, \dots, m_L\}$ of trailers are available. $K_1^T = \{1, \dots, m_L\}$ and $K_2^T = \{m_L + 1, \dots, m_T\}$ are the subsets of K^T that consist of trucks that can pull a trailer and pure trucks (without trailer attached), respectively. Note that $|K_1^T| = |K^L|$, that is, there are m_L complete vehicles. In addition, m_L is the number of trailers, and m_T is the number of trucks ($m_L \leq m_T$). Complete vehicles and pure trucks are assumed to be homogeneous. Let Q_T be the capacity of each truck and Q_L be the capacity of each trailer. Hence, $Q_T + Q_L$ is the capacity of a complete vehicle.

As mentioned above and illustrated in Fig. 1, three different types of routes can appear in this variant of the TTRP. For MVRs, the complete vehicle leaves the depot and serves some VCs; this part of the MVR is known as the main tour. The main tour is entirely covered by a complete vehicle and starts and ends at the depot. During the tour of an MVR, it is possible to uncouple the trailer from the truck and leave it parked at one of the VC locations to start a sub-tour (or even at the depot, which is always a candidate for trailer parking places). VCs and TCs can be served in a sub-tour because they are performed by a pure truck. Sub-tours begin and end at the parking place (the depot or any of the VCs of the main tour), also known as the root of the sub-tour. There are no restrictions on the number of sub-tours in an MVR or on the number of sub-tours that can start from the same VC on a given main tour, as long as the vehicle capacity restrictions are satisfied. That is, the demands transported on the MVR cannot exceed the capacity of a complete vehicle, $Q_T + Q_L$, and Q_T cannot be surpassed in a sub-tour. Another type of route is a PVR, which is fully traveled by a complete vehicle, implying that only VCs can be delivered to and their demands cannot exceed $Q_T + Q_L$. On the contrary, PTRs serve both types of customers because trucks travel without a trailer attached. The demand transported on a PTR cannot exceed Q_T.

For the sake of simplicity, it is also assumed that travel costs are the same for all vehicles, regardless of whether a trailer is attached. Each trailer $r \in K^L$ is divided into a set of compartments, H^L, where Q_L^H is the capacity of a trailer compartment. Similarly, each truck $k \in K^T$ is split into a set of compartments, H^T, where Q_T^H is the capacity of a truck compartment. Furthermore, let the set $F = \{1, \ldots, n_F\}$ of feed types be given. Each node (except for the depot) has a nonnegative demand d_{if} $(i \in N\backslash\{0\}, f \in F)$ for every feed type. The demands must be served at customers' locations and transported from the depot without the feed types being mixed. In addition, products for different customers cannot be carried within the same compartment. The total demands of each customer must be met by the same vehicle, and it is possible to divide a customer's demand for the same feed type among several compartments. Trucks and trailers have a maximum usage time allowed of D, an average speed of vm, and legal capacities, L_T and L_L, respectively, which may appear depending on regulations or laws in some specific areas.

Regarding the decision variables involved in the model, x_{ij}^{kr} and y_{ij}^{klv} (both binary) are related to the construction of routes. The former involves routes covered by complete vehicles (MVRs or PVRs), and it takes a value of 1 if the complete vehicle consisting of truck $k \in K_1^T$ and trailer $r \in K^L$ travels from node i to j $(i, j \in N_1 \cup \{0\})$; otherwise, it is 0. In contrast, y_{ij}^{klv} refers to routes covered only by trucks (PTRs or sub-tours of MVRs), and it takes a value of 1 if truck $k \in K^T$ traverses the arc (i, j) on the vth route/sub-tour ($v \in \mathcal{V} = \{1, \ldots, n\}$)[1] with root $l \in N_1 \cup \{0\}$; otherwise, it is 0. For $l = 0$, the associated tour is a PTR ($k \in K_2^T$) or a sub-tour in an MVR whose root is the depot ($k \in K_1^T$). However, if $l \in N_1$, then such a root refers to the VC of the main tour of an MVR working as a trailer parking place to start a sub-tour (and $k \in K_1^T$). The remaining variables are related to the vehicle compartments. In particular, $ZT_{i,f,ht}^k$ takes values in [0,1] and represents the proportion of compartment $ht \in H^T$ of truck $k \in K^T$ carrying feed $f \in F$ for customer $i \in N$, whereas $U_{i,f,ht}^k$ is a binary variable equal to 1 if $ZT_{i,f,ht}^k > 0$ and 0 otherwise. Analogously, $ZL_{i,f,hl}^r$ takes values in [0,1] representing the proportion of compartment $hl \in H^L$ of trailer $r \in K^L$ loaded with feed $f \in F$ for customer $i \in N_1$, while $V_{i,f,hl}^r$ is a binary variable equal to 1 if $ZL_{i,f,hl}^r > 0$ and 0 otherwise.

The objective of the MC-TTRP is to determine a set of vehicle tours that minimizes the total cost of all edges to be traveled, that is, the total distance of the solution routes, such that all constraints are met, that is, the demands are satisfied, no vehicle capacities are exceeded, and the restrictions of access to customers and constraints related to the loading of compartments are considered.

From now on, we will denote the set of VCs and the depot, $N_1 \cup \{0\}$, as N_1^0, and we will denote the set of customers, $N \setminus \{0\}$, by N^*. Table 1 gives a summary of the sets, parameters, and decision variables involved in the model to facilitate better understanding of our proposal. Given this termi-

[1]Note that a root candidate $l \in N_1 \cup \{0\}$ can have as many sub-tours as there are customers, n.

Table 1
Notation of the proposed MC-TTRP

Set or parameter	Definition	Parameter	Definition
$\{0\}$	Depot	n	Total number of customers
N_1	Set of VCs	n_1	Number of VCs
N_2	Set of TCs	m_L	Number of trailers (and of complete vehicles)
$N = \{0\} \cup N_1 \cup N_2$	Set of nodes (customers and depot)	m_T	Number of trucks
$N^* = N \setminus \{0\}$	Set of customers	c_{ij}	Distance a vehicle must travel from i to j
$N_1^0 = N_1 \cup \{0\}$	Set of VCs and depot	n_F	Number of different types of feed
K_1^T	Set of trucks that can hitch a trailer	d_{if}	Demand of customer i for feed f
K_2^T	Set of pure trucks	Q_T	Capacity of a truck
$K^T = K_1^T \cup K_2^T$	Set of trucks	Q_L	Capacity of a trailer
K^L	Set of trailers	Q_T^H	Capacity of a truck's hopper
F	Set of different types of feed	Q_L^H	Capacity of a trailer's hopper
H^T	Set of truck hoppers	D	Maximum time allowed to use a truck/vehicle
H^L	Set of trailer hoppers	vm	Average speed of trucks/vehicles
$\mathcal{V}$	Set of tours/sub-tours leaving a specific root	s_i	Service time required for node $i \in N$
L_T	Legal capacity of trucks	L_L	Legal capacity of trailers

Variable	Definition
x_{ij}^{kr}	Binary variable equal to 1 if truck k with trailer r passes through arc (i, j); 0 otherwise
y_{ij}^{klv}	Binary variable equal to 1 if truck k passes through arc (i, j) on route/sub-tour v with parking place l; 0 otherwise
$U_{i,f,ht}^k$	Binary variable equal to 1 if compartment ht of truck k is loaded with feed f for customer i; 0 otherwise
$V_{i,f,hl}^r$	Binary variable equal to 1 if compartment hl of trailer r is loaded with feed f for customer i; 0 otherwise
$ZT_{i,f,ht}^k$	Proportion of compartment ht of truck k loaded with feed f for customer i
$ZL_{i,f,hl}^r$	Proportion of compartment hl of trailer r loaded with feed f for customer i

nology and notation, the objective function and constraints of the MC-TTRP can be formulated as follows:

minimize:

$$\sum_{i\in N_1^0}\sum_{j\in N_1^0}\sum_{k\in K_1^T}\sum_{r\in K^L} c_{ij}x_{ij}^{kr} + \sum_{i\in N}\sum_{j\in N}\sum_{k\in K^T}\sum_{l\in N_1^0}\sum_{v\in\mathcal{V}} c_{ij}y_{ij}^{klv} \qquad (1)$$

subject to

$$\sum_{i\in N_1^0}\sum_{k\in K_1^T}\sum_{r\in K^L} x_{ij}^{kr} + \sum_{i\in N}\sum_{k\in K^T}\sum_{\substack{l\in N_1^0\\ l\neq j}}\sum_{v\in\mathcal{V}} y_{ij}^{klv} = 1, \quad j\in N_1; \qquad (2)$$

$$\sum_{i\in N_1^0}\sum_{k\in K_1^T}\sum_{r\in K^L} x_{ij}^{kr} + \sum_{i\in N}\sum_{k\in K^T} y_{ij}^{kjv} \leq 2, \quad j\in N_1;\ v\in\mathcal{V}; \qquad (3)$$

$$\sum_{i\in N}\sum_{k\in K^T}\sum_{l\in N_1^0}\sum_{v\in\mathcal{V}} y_{ij}^{klv} = 1, \quad j \in N_2; \tag{4}$$

$$\sum_{i\in N}\sum_{k\in K^T}\sum_{l\in N_1}\sum_{v\in\mathcal{V}} y_{i0}^{klv} = 0, \tag{5}$$

$$\sum_{j\in N} y_{lj}^{klv} \leq 1, \quad k \in K^T;\ l \in N_1^0;\ v \in \mathcal{V}; \tag{6}$$

$$y_{lj}^{klv} \leq \sum_{i\in N_1^0}\sum_{r\in K^L} x_{il}^{kr}, \quad j \in N^*;\ k \in K_1^T;\ l \in N_1;\ v \in \mathcal{V}; \tag{7}$$

$$y_{ij}^{klv} \leq \sum_{p\in N} y_{lp}^{klv}, \quad i, j \in N;\ k \in K^T;\ l \in N_1^0;\ v \in \mathcal{V}; \tag{8}$$

$$\sum_{i\in N}\sum_{j\in N}\sum_{v\in\mathcal{V}} y_{ij}^{klv} = 0, \quad k \in K_2^T;\ l \in N_1; \tag{9}$$

$$\sum_{i\in N}\sum_{j\in N}\sum_{v\in\mathcal{V}} y_{ij}^{k0v} = 0, \quad k \in K_1^T; \tag{10}$$

$$\sum_{v\in\mathcal{V}}\sum_{j\in N^*} y_{0j}^{k0v} \leq 1, \quad k \in K_2^T; \tag{11}$$

$$\sum_{j\in N_1}\sum_{r\in K^L} x_{0j}^{kr} \leq 1, \quad k \in K_1^T; \tag{12}$$

$$\sum_{j\in N_1}\sum_{k\in K_1^T} x_{0j}^{kr} \leq 1, \quad r \in K^L; \tag{13}$$

$$\sum_{i\in N^*}\sum_{\substack{j\in N^*\\ j\neq l}}\sum_{f\in F} d_{jf} y_{ij}^{klv} \leq Q_T, \quad k \in K_1^T;\ l \in N_1;\ v \in \mathcal{V}; \tag{14}$$

$$\sum_{i\in N}\sum_{j\in N^*}\sum_{f\in F} d_{jf} y_{ij}^{k0v} \leq Q_T, \quad k \in K_2^T;\ v \in \mathcal{V}; \tag{15}$$

$$\sum_{i\in N_1^0}\sum_{j\in N_1}\sum_{f\in F} d_{jf} x_{ij}^{kr} + \sum_{i\in N^*}\sum_{\substack{j\in N^*\\ j\neq l}}\sum_{l\in N_1}\sum_{v\in\mathcal{V}}\sum_{f\in F} d_{jf} y_{ij}^{klv} \leq Q_T + Q_L, \quad k \in K_1^T;\ r \in K^L; \tag{16}$$

$$\sum_{i\in N_1}\sum_{j\in N_1^0} s_j x_{ij}^{kr} + \sum_{l\in N_1^0}\sum_{v\in\mathcal{V}}\sum_{\substack{i\in N^*\\ i\neq l}}\sum_{j\in N} s_j y_{ij}^{klv} + \sum_{i\in N_1^0}\sum_{j\in N_1^0}(c_{ij}/vm)x_{ij}^{kr}$$

$$+\sum_{l\in N_1^0}\sum_{v\in\mathcal{V}}\sum_{i\in N}\sum_{j\in N}(c_{ij}/vm)y_{ij}^{klv} \le D - s_0, \quad k\in K_1^T;\ r\in K^L; \tag{17}$$

$$\sum_{v\in\mathcal{V}}\sum_{i\in N}\sum_{j\in N}(s_j + c_{ij}/vm)y_{ij}^{k0v} \le D, \quad k\in K_2^T; \tag{18}$$

$$\sum_{i\in N} y_{ij}^{klv} = \sum_{p\in N} y_{jp}^{klv}, \quad j\in N;\ k\in K^T;\ l\in N_1^0;\ v\in\mathcal{V}; \tag{19}$$

$$\sum_{i\in N_1^0} x_{ij}^{kr} = \sum_{p\in N_1^0} x_{jp}^{kr}, \quad j\in N_1^0;\ k\in K_1^T;\ r\in K^L; \tag{20}$$

$$\sum_{i\in B}\sum_{j\in B} x_{ij}^{kr} \le |B|-1, \quad k\in K_1^T;\ r\in K^L;\ \forall B\subseteq N_1 : |B|\ge 2; \tag{21}$$

$$\sum_{i\in B}\sum_{j\in B} y_{ij}^{klv} - \sum_{i\in B\cap N_1}\sum_{j\in N_1\setminus B}\sum_{r\in K^L} x_{ij}^{kr} \le |B|-1, \quad k\in K_1^T;\ l\in N_1;$$

$$v\in\mathcal{V};\ \forall B\subseteq N : |B|\ge 2; \tag{22}$$

$$\sum_{i\in B}\sum_{j\in B} y_{ij}^{k0v} \le |B|-1, \quad k\in K_2^T;\ v\in\mathcal{V};\ \forall B\subseteq N : |B|\ge 2; \tag{23}$$

$$\frac{1}{|H^T|}\sum_{f\in F}\sum_{ht\in H^T} ZT_{j,f,ht}^k \le \sum_{i\in N_1^0}\sum_{r\in K^L} x_{ij}^{kr} + \sum_{i\in N}\sum_{l\in N_1}\sum_{v\in\mathcal{V}} y_{ij}^{klv}, \quad k\in K_1^T;\ j\in N_1; \tag{24}$$

$$\frac{1}{|H^T|}\sum_{f\in F}\sum_{ht\in H^T} ZT_{j,f,ht}^k \le \sum_{i\in N}\sum_{v\in\mathcal{V}} y_{ij}^{k0v}, \quad k\in K_2^T;\ j\in N_1; \tag{25}$$

$$\frac{1}{|H^L|}\sum_{f\in F}\sum_{hl\in H^L} ZL_{j,f,hl}^r \le \sum_{i\in N_1^0}\sum_{k\in K_1^T} x_{ij}^{kr}, \quad r\in K^L;\ j\in N_1; \tag{26}$$

$$\frac{1}{|H^T|}\sum_{f\in F}\sum_{ht\in H^T} ZT_{j,f,ht}^k \le \sum_{i\in N}\sum_{l\in N_1^0}\sum_{v\in\mathcal{V}} y_{ij}^{klv}, \quad k\in K^T;\ j\in N_2; \tag{27}$$

$$\sum_{i\in N_1^0} x_{ij}^{kr} \le \sum_{f\in F}\sum_{ht\in H^T} Q_T^H ZT_{j,f,ht}^k + \sum_{f\in F}\sum_{hl\in H^L} Q_L^H ZL_{j,f,hl}^r, \quad k\in K_1^T;\ r\in K^L;$$

$$j\in N_1; \tag{28}$$

$$\sum_{i\in N}\sum_{v\in\mathcal{V}} y_{ij}^{k0v} \le \sum_{f\in F}\sum_{ht\in H^T} Q_T^H ZT_{j,f,ht}^k, \quad k\in K_2^T;\ j\in N_1; \tag{29}$$

$$\sum_{i\in N}\sum_{l\in N_1}\sum_{v\in\mathcal{V}} y_{ij}^{klv} \le \sum_{f\in F}\sum_{ht\in H^T} Q_T^H ZT_{j,f,ht}^k, \quad k\in K_1^T;\ j\in N_1; \tag{30}$$

$$\sum_{i\in N}\sum_{l\in N_1^0}\sum_{v\in\mathcal{V}} y_{ij}^{klv} \le \sum_{f\in F}\sum_{ht\in H^T} Q_T^H ZT_{j,f,ht}^k, \quad k\in K^T;\ j\in N_2; \tag{31}$$

$$\sum_{r\in K^L}\sum_{hl\in H^L} Q_L^H ZL_{j,f,hl}^r + \sum_{k\in K^T}\sum_{ht\in H^T} Q_T^H ZT_{j,f,ht}^k = d_{jf}, \quad j\in N_1;\ f\in F; \tag{32}$$

$$\sum_{k\in K^T}\sum_{ht\in H^T} Q_T^H ZT_{j,f,ht}^k = d_{jf}, \quad j\in N_2;\ f\in F; \tag{33}$$

$$\sum_{i\in N^*}\sum_{f\in F}\sum_{ht\in H^T} Q_T^H ZT_{i,f,ht}^k \le L_T, \quad k\in K^T; \tag{34}$$

$$\sum_{i\in N_1}\sum_{f\in F}\sum_{hl\in H^L} Q_L^H ZL_{i,f,hl}^r \le L_L, \quad r\in K^L; \tag{35}$$

$$\sum_{i\in N^*}\sum_{f\in F} ZT_{i,f,ht}^k \le 1, \quad k\in K^T;\ ht\in H^T; \tag{36}$$

$$\sum_{i\in N_1}\sum_{f\in F} ZL_{i,f,hl}^r \le 1, \quad r\in K^L;\ hl\in H^L; \tag{37}$$

$$ZT_{i,f,ht}^k - U_{i,f,ht}^k \le 0, \quad i\in N^*;\ f\in F;\ k\in K^T;\ ht\in H^T; \tag{38}$$

$$ZL_{i,f,hl}^r - V_{i,f,hl}^r \le 0, \quad i\in N_1;\ f\in F;\ r\in K^L;\ hl\in H^L; \tag{39}$$

$$U_{i,f_1,ht}^k + U_{j,f_2,ht}^k \le 1, \quad i,j\in N^*;\ f_1,f_2\in F;\ f_1\ne f_2;\ k\in K^T;\ ht\in H^T; \tag{40}$$

$$V_{i,f_1,hl}^r + V_{j,f_2,hl}^r \le 1, \quad i,j\in N_1;\ f_1,f_2\in F;\ f_1\ne f_2;\ r\in K^L;\ hl\in H^L; \tag{41}$$

$$U_{i,f,ht}^k + U_{j,f,ht}^k \le 1, \quad i,j\in N^*;\ i\ne j;\ f\in F;\ k\in K^T;\ ht\in H^T; \tag{42}$$

$$V_{i,f,hl}^r + V_{j,f,hl}^r \le 1, \quad i,j\in N_1;\ i\ne j;\ f\in F;\ r\in K^L;\ hl\in H^L; \tag{43}$$

$$x_{ij}^{kr} \in \{0,1\}, \quad i,j\in N_1^0;\ k\in K_1^T;\ r\in K^L; \tag{44}$$

$$y_{ij}^{klv} \in \{0,1\}, \quad i,j\in N;\ k\in K^T;\ l\in N_1^0;\ v\in\mathcal{V}; \tag{45}$$

$$ZT_{i,f,ht}^k \in [0,1], \quad i\in N;\ f\in F;\ k\in K^T;\ ht\in H^T; \tag{46}$$

$$ZL^r_{i,f,hl} \in [0, 1], \quad i \in N_1^0;\ f \in F;\ r \in K^L;\ hl \in H^L; \tag{47}$$

$$U^k_{i,f,ht} \in \{0, 1\}, \quad i \in N;\ f \in F;\ k \in K^T;\ ht \in H^T; \tag{48}$$

$$V^r_{i,f,hl} \in \{0, 1\}, \quad i \in N_1^0;\ f \in F;\ r \in K^L;\ hl \in H^L; \tag{49}$$

The objective function and above restrictions are explained in the following, separating them into thematic blocks to fully understand the model:

- **Objective function (1)**: This minimizes the total cost of all tours. The first term refers to the routes traveled by a complete vehicle (PVRs and main routes of MVRs), while the second term includes the PTRs and MVR sub-tours.
- **Customer-specific restrictions (2–8)**: (2) establishes that each VC must be present exactly once either in the main route of a complete route or in a sub-tour of which it is not the parking place. From (3), a VC can be present twice if it is the root of a sub-tour. (4) indicates that TCs must be visited only once, either on a sub-tour or on a PTR (when $l = 0$). From (5), the depot cannot be present in any sub-tour that begins at a VC. Constraint (6) shows that for each sub-tour or PTR, the corresponding truck goes from the parking place or depot, as appropriate, to a customer no more than once. (7) implies that if a VC is not served by a complete vehicle, then that customer cannot be a parking place candidate to start a sub-tour. (8) describes that other customers can only belong to a sub-tour starting from a candidate parking place if that candidate is actually selected as a parking place.
- **Vehicle-specific restrictions (9–18)**: From (9), trucks that do not have an associated trailer do not have a main route. (10) indicates that trucks with a trailer attached cannot perform PTRs. From (11), for trucks without trailers, there can be at most one route, that is, a PTR (there is no multiple use of vehicles). From constraints (12–13), for complete vehicles, the number of sub-tours is not limited, but the number of main routes cannot exceed one. (14–15) describe the customer demands, which cannot exceed the capacity of a truck, Q_T, on a route/sub-tour without a trailer. (16) explains the demand limits on a route performed by a truck pulling a trailer, which cannot exceed $Q_T + Q_L$. From (17), complete vehicles cannot exceed their maximum usage time, including the loading or unloading time of a vehicle in the depot, s_0. (18) specifies the maximum usage time in the case of trucks, which also cannot be surpassed.
- **Formulation-specific restrictions (19–23)**: (19) covers flow conservation in sub-tours and PTRs. (20) represents the flow conservation on PVRs and main routes. (21) provides disconnected cycle elimination constraints on the main routes of MVRs and PVRs. (22) outlines the suppression of disconnected cycles in sub-tours. (23) models the removal of disconnected cycles on PTRs.
- **Relations between routes and load distribution in different compartments (24–31)**: (24) specifies that if a truck in an MVR or PVR does not visit a VC, then that truck does not load goods for that customer in any of its compartments. From (25), if a truck in a PTR does not visit a VC, then that truck does not load products for that customer in any of its compartments. From (26), if a trailer does not visit a VC, then that trailer does not load goods for that customer in any of its compartments. From (27), if a truck does not visit a TC, then it does not load goods for that customer. (28) states that if a complete vehicle visits a VC, then either the truck or the trailer is loaded with goods for that customer. From (29), if a VC is visited by a truck (in a PTR), then

the truck distributes goods for that customer. Constraint (30) states that if a truck in a sub-tour serves a VC, then that truck is loaded with goods for that customer. From (31), if a truck visits a TC, then it transports goods to that customer in some of its compartments.

- **Demand delivery restrictions (32–33)**: From (32), every VC receives all of its demand. From constraint (33), every TC is delivered all of its demand.
- **Volume of goods that can be transported (34–37)**: From (34), no truck loads more than what is legally allowed. Constraint (35) states that no trailer loads more than what is legally allowed. (36) states that no truck compartment loads above its capacity. From (37), no trailer compartment loads above its capacity.
- **Technical restrictions in the loading procedure (38–43)**: (38) defines the logical relationship between U and ZT. (39) defines the logical relationship between V and ZL. From (40), it is not possible to mix products of different types (neither from the same customer nor from different customers) in the same compartment of a truck. Constraint (41) states that products of different types cannot be mixed (neither from the same customer nor from different customers) in the same compartment of a trailer. (42) prohibits mixing products from different customers in the same truck compartment. (43) disallows the mixing of products from different customers in the same compartment of a trailer.
- **Nature of the variables involved in the model (44–49)**: From (44) and (45), x and y variables are binary, respectively. (46) and (47) specify that ZT and ZL variables take values in [0,1], respectively. From (48) and (49), U and V variables are binary, respectively.

Our proposed MC-TTRP offers a comprehensive way to model compartments using the TTRP approach. It was formulated as an MILP problem, considering the complexity associated with this type of problem. The following sections analyze how to solve it both approximately or exactly, alongside the advantages and disadvantages of each method.

4. Heuristic approach

Our proposal for solving the novel MC-TTRP model is a two-phase method: (1) a constructive heuristic that creates a feasible solution and (2) a metaheuristic approach that iteratively improves the solution provided in the previous stage. Moreover, our algorithm can consider additional features, such as fleet limitations, and solve the classic TTRP. The following subsections describe the proposed method in detail.

4.1. Construction phase

For the first phase, we designed an *ad hoc* Clarke–Wright algorithm (CW) capable of handling TTRP problems with compartments. The CW is a popular constructive heuristic for the basic VRP. Its strategy is to build a feasible solution based on the notion of savings in the routing cost.

Typically, the CW begins by creating a starting solution for which all routes start at the depot, visit one customer, and return to the depot. It continues by computing the savings of joining each pair of these routes. Throughout each iteration, the CW considers the savings in descending order

Algorithm 1. Savings-based construction heuristic for the MC-TTRP

procedure CW_MC-TTRP ($n, n_1, C, d, Q_T, Q_L, H^T, H^L, f$)
  Create $N_1 = \{1, \ldots, n_1\}$ and $N_2 = \{n_1 + 1, \ldots, n\}$ ▷ Initialization.
  Create R and $\hat{R}$
  Create S and $\hat{S}$
  Create matrices H^T_{rem}, H^L_{rem}, and T
  $S_m = 1$
  while $S_m > 0$ **do** ▷ Constructive heuristics.
    S_m = Maximum saving in $\{S, \hat{S}\}$
    if $S_m \in S$ **then**
      Extract coordinates $[i, j]$ of $S_m \in S$
      Check type of routes r_i and r_j using T
      if r_i and r_j can be merged **then**
        Apply fusion ▷ Different fusion cases can be seen in the description.
        Update $\{R, \hat{R}, S, \hat{S}, H^L_{\text{rem}}, H^T_{\text{rem}}, T\}$
      else
        Update $\{S, \hat{S}\}$
      end if
    else
      Extract coordinates $[i, j]$ of $S_m \in \hat{S}$
      l = Check last customer using r_i or r_j and R
      if r_i or r_j can be converted into a sub-tour **then**
        Create MVR
        Update $\{R, \hat{R}, S, \hat{S}, H^L_{\text{rem}}, H^T_{\text{rem}}, T\}$
      else
        Update $\{\hat{S}\}$
      end if
    end if
  end while
  `solution_route` = Create route using R, $\hat{R}$, and T
  Add disconnected customers in `solution_route`, and update $\{T, H^T_{\text{rem}}, H^L_{\text{rem}}\}$ ▷ Refining the construction phase.
  Adjust vehicle fleet creating or completing MVRs in `solution_route`, and update $\{R, \hat{R}, T, H^L_{\text{rem}}, H^T_{\text{rem}}\}$
  Adjust vehicle fleet splitting routes in `solution_route`, and update $\{R, \hat{R}, T, H^L_{\text{rem}}, H^T_{\text{rem}}\}$
  Adjust vehicle fleet switching residual routes in `solution_route`, and update $\{R, \hat{R}, T, H^L_{\text{rem}}, H^T_{\text{rem}}\}$
  Apply tour improvement in `solution_route`, and update $\{R, \hat{R}, T\}$
  Apply local search movements in `solution_route`, and update $\{R, \hat{R}, T, H^L_{\text{rem}}, H^T_{\text{rem}}\}$
  `cw_solution` = Create final route using R, $\hat{R}$, and `solution_route`
  return(list(`cw_solution`, T))
end procedure

to choose which routes to merge, given that such merging is feasible. It stops when all customers have been served.

The pseudocode for our proposed CW is shown in Algorithm 1. First, the initial parameters, such as the total number of customers and number of VCs (n and n_1, respectively), matrices of demands and distances (d and C), vehicle capacities (Q_T and Q_L), vehicle compartment configuration (H^T and H^L), and number of different types of products (f), are defined. Moreover, unlike classic CW, our heuristic considers two matrices of routes: R and $\hat{R}$. The former is updated considering

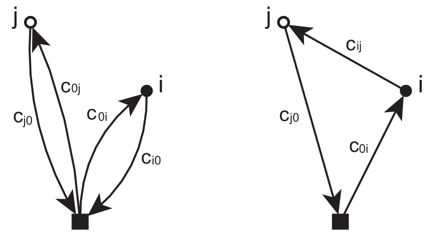


Fig. 2. Calculation of s_{ij} for $i, j \in N$.

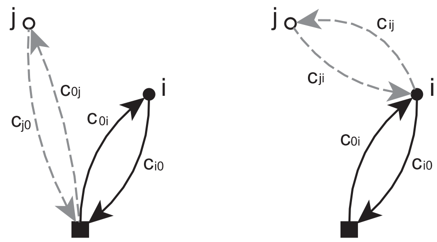


Fig. 3. Calculation of $\hat{s}_{ij}$ for $i \in N_1$ and $j \in N_2$.

those tours directly connected to the depot, while $\hat{R}$ considers the sub-tours whose roots are VCs. Consequently, as indicated in line 3 of Algorithm 1, we initialize both matrices as follows:

$$R = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 2 & 0 \\ \vdots & \vdots & \vdots \\ 0 & n & 0 \end{pmatrix} = \hat{R},$$

assuming that routes $(0, i, 0)$ for $i \in N_1$ are traveled by complete vehicles and routes $(0, j, 0)$ for $j \in N_2$ are covered by trucks alone. Each customer's corresponding row represents the previous and next customer in the transport network, as appropriate.

Furthermore, at the beginning of the proposed algorithm, two saving matrices are created: S and $\hat{S}$. The entries of S are computed as $s_{ij} = c_{i0} + c_{0j} - c_{ij}$ (for $i, j \in N, i \neq j$)[2], which represents the standard savings matrix. This scenario is illustrated in Fig. 2. However, because we have two different classes of customers and given that VCs can serve as trailer parking places, we also consider savings $\hat{s}_{ij}$. These values are calculated as follows:

- $\hat{s}_{ij} = s_{ij}$ for $i, j \in N_1$ or $i, j \in N_2$.
- $\hat{s}_{ij} = c_{j0} + c_{0j} - c_{ij} - c_{ji}$ for $i \in N_1$ and $j \in N_2$. These savings are the result of removing route $(0, j, 0)$ and converting it into a sub-tour whose root is i. That is, we will have the route $(0, \ldots, i, j, i, \ldots, 0)$ instead of $(0, j, 0)$ and $(0, \ldots, i, \ldots, 0)$. Figure 3 illustrates this situation.
- $\hat{s}_{ij} = c_{i0} + c_{0i} - c_{ij} - c_{ji}$ for $j \in N_1$ and $i \in N_2$. These savings are analogous to the previous ones, swapping i and j in their type of customer.

[2]Moreover, $s_{ii} = 0$ for all $i \in N$.

Hence, matrix $\hat{S}$ will have the following structure:

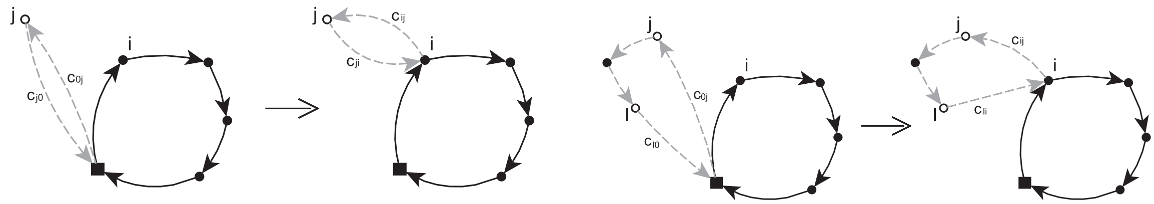
$$\hat{S} = \left(\begin{array}{ccc|ccc} s_{1,1} & \cdots & s_{1,n_1} & \hat{s}_{1,n_1+1} & \cdots & \hat{s}_{1,n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ s_{n_1,1} & \cdots & s_{n_1,n_1} & \hat{s}_{n_1,n_1+1} & \cdots & \hat{s}_{n_1,n} \\ \hline \hat{s}_{n_1+1,1} & \cdots & \hat{s}_{n_1+1,n_1} & s_{n_1+1,n_1+1} & \cdots & s_{n_1+1,n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \hat{s}_{n,1} & \cdots & \hat{s}_{n,n_1} & s_{n,n_1+1} & \cdots & s_{n,n} \end{array}\right).$$

We must also consider the features related to compartments. As shown in line 5 of Algorithm 1, we create matrices H^T_{rem}, H^L_{rem}, and T. The former two are initialized as H^T and H^L, respectively, and are updated by setting the compartments that have already been used to -1. Matrix T plays a fundamental role in this implementation because it contains all information about the vehicles' loading procedures; it indicates the allocation details of each customer's demand for its corresponding vehicle.

Once all initial parameters have been created, our CW implementation follows an iterative process, where different routes are merged based on the maximum saving, S_m (lines 7–28 of Algorithm 1). Thereafter, r_i and r_j are considered to be routes containing customers i and j, respectively. By checking matrix T (line 11), the algorithm knows which vehicles cover each of these routes; thus, we can determine the PTRs, PVRs, and MVRs. Therefore, our proposed method repeats a set of steps, while S_m, the maximum saving of S and $\hat{S}$, is positive (stop condition). Thereafter, each iteration of the algorithm has two discernible cases: $S_m \in S$ or its opposite, $S_m \in \hat{S}$.

The first case occurs when S_m belongs to S (lines 9–17 of the pseudocode). Our heuristic applies the classic routing merge of the CW, which considers the type of customers (VCs or TCs) involved in such saving, which is a decisive factor in the type of route generated. Thus, four different unions between routes (by connecting customer i to customer j) can be conducted, as long as i is the first customer of route r_i and j the last one of route r_j:

1. **Merging two PVRs**: If routes r_i and r_j are PVRs covered by different complete vehicles, where the trucks are not yet loaded, the algorithm must count the number of unavailable compartments between both trailers. Two situations arise from this: (i) If the number does not exceed the number of trailer compartments, our method can move all goods to one trailer, emptying the other. (ii) If the number exceeds the number of trailer compartments, our method starts to load one of the trucks as long as the total number of available compartments is sufficient to meet the demands of both routes. Thus, it moves the goods from the other trailer to the chosen complete vehicle, giving preference to the filling of the trailer.
2. **Merging two PTRs**: When both routes r_i and r_j are PTRs, our heuristic can merge them as long as the number of unavailable compartments between both trucks does not surpass the number of truck compartments. In this case, the goods are moved to one of the trucks, leaving the other free.
3. **Combining an MVR with a PVR**: If one of the routes is an MVR and the other is a PVR with an empty truck, say r_i and r_j, respectively, then their union is possible when the demands already



(a) Before (left) and after (right) merging when r_j has a single customer.

(b) Before (left) and after (right) merging when r_j has multiple customer.

Fig. 4. Merges performed when $S \in S_m$.

met by route r_j fit in the trailer of route r_i. That is, when the trailer of route r_i has a sufficient number of available compartments to move goods from the trailer of route r_j.

4. **Inserting route r_i in a PTR**: If route $r_i = (0, i, 0)$, where i is a VC, and route r_j is a PTR, then merging both routes is possible when the number of available compartments in the truck is sufficient to hold i's demands.

For all the above considered fusions, updating matrices $\{R, \hat{R}, S, \hat{S}, H^T_{\text{rem}}, H^L_{\text{rem}}, T\}$ is necessary to contemplate the changes conducted, as indicated in line 14. If merging cannot be completed, we must simply update matrices S and $\hat{S}$ by setting their corresponding entries to zero.

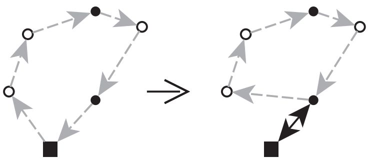
The second case occurs when $\hat{S}$ contains S_m, which means that the current saving arises from merging a PTR and PVR (obtaining a sub-tour as a result). Lines 18–27 of Algorithm 1 correspond to this situation. In such a case, it is clear that customers involved in S_m have different types. Let us see how to proceed when $i \in N_1$ and $j \in N_2$, because the other case is analogous. Our method can only execute merges when r_i is a PVR with an empty truck and r_j is a PTR. With this in mind, two possibilities arise:

- If $r_j = (0, j, 0)$, the algorithm uncouples the truck of route r_i and hitches the truck of route r_j, loaded with j's demands, to the trailer of route r_i. This proceeds as illustrated in Fig. 4a.
- Suppose j is the first customer of the route but not the only one. In this case, we must also consider the last customer of route r_j, say, l. To join routes r_i and r_j, it is necessary that $c_{0j} + c_{l0} - c_{ij} - c_{li}$ is positive. We illustrate this merging in Fig. 4b.

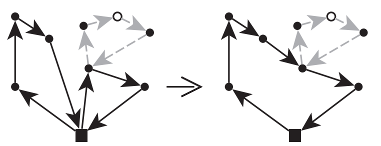
4.2. Refining the construction phase

Once we have exited the main loop and no saving is positive, the algorithm must check whether the resulting route configuration is feasible. The following post-processing functions (lines 30–35 of the pseudocode) are applied in the order in which they are presented to verify such feasibility, correct the tours if needed, improve the quality of the results, and limit the vehicle fleet:

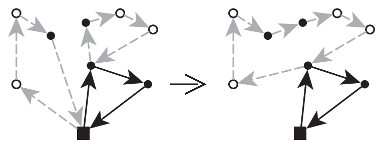
- **Add disconnected customers**: This checks if there exists a route $r_i = (0, i, 0)$, which means that customer i has not yet been served. During the initialization of R, we mentioned that the starting routes should be covered by complete vehicles or trucks alone, depending on whether i is a VC or



(a) Convert a PTR into an MVR.



(b) Merge a PVR into the main tour of an MVR.



(c) Merge a PTR into the sub-tour of an MVR.

Fig. 5. Adjusting vehicle fleet by creating or completing MVRs.

TC, respectively. Considering this, we supply customer i with its corresponding type of vehicle, contemplating the compartment load in matrix T. After that, we are able to add the route $(0, i, 0)$ to the solution.

- **Adjust vehicle fleet by creating or completing MVRs**: The aim of this function is to use all the available trailers in the solution obtained by eliminating excess PTRs and PVRs, either by changing or merging them into MVRs, respectively. First, because all PTRs with at least a VC can be transformed into MVRs, we convert as many PTRs into MVRs as necessary to reach the required number of trailers. Subsequently, the lowest quality PVRs and PTRs are selected to be joined either to the main tour or to one of the sub-tours of an existing MVR, provided that the result is a feasible solution. This procedure is illustrated in Fig. 5.
- **Adjust vehicle fleet by destroying routes**: It is assumed that the routes to be deleted or preserved have already been selected to adjust the fleet. Iteratively, an attempt is made to insert the first group into the second group. The method of merging these routes depends on their nature. For instance, a PTR could only be part of a sub-tour of an MVR or join another PTR. If there are no feasible insertions of a specific route to be deleted, it is split in half. The algorithm will try to add it again in the next iteration, repeating this process until there are no more residual routes or until they cannot be divided anymore.
- **Adjust vehicle fleet by switching residual routes**: After applying the previous function, if there are still routes to be eliminated, they will consist of a single customer. This last procedure to adjust the fleet exchanges these residual customers for others that have less load on the routes to be preserved while aiming to keep the cost function from increasing as much as possible. Once the exchange is conducted, the customer to be inserted requires less space. In this manner, the algorithm aims to relocate it to one of the existing routes, repeating the process until there are no residual routes left.
- **Apply tour improvement**: This function performs 2-opt, 3-opt, and 4-opt* moves on the resulting routes[3], including the sub-tours, individually. For each route, the move that leads to the greatest cost reduction, if any, is applied.
- **Apply local search moves**: Finally, a local search is applied to the solution obtained from Algorithm 1. This iteratively applies a set of small modifications to intensify the search on routes close to the current solution. To implement these moves, we were inspired by the work of Derigs et al. (2013), where a hybrid approach for solving the TTRP, combining local search and large neighborhood search, was presented. We adapted some of the moves implemented in their local

[3]The 4-opt* procedure uses a subset of potential 4-opt moves, cf. Renaud et al. (1996).

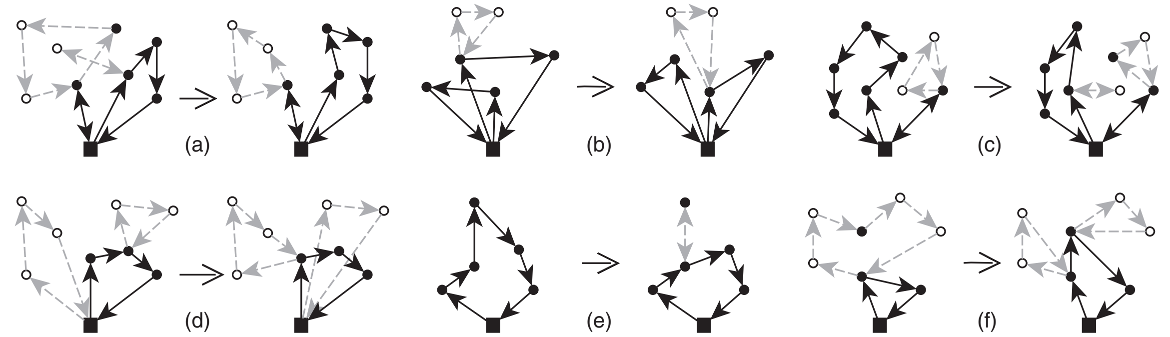


Fig. 6. Local search moves for the MC-TTRP. (a) Replacing the single customer of a sub-tour with a VC, (b) replacing a sub-tour root with another VC, (c) replacing a main tour customer with a TC, (d) exchanging a PTR and a sub-tour and relocating the parking place, (e) moving a main tour customer to a new sub-tour, and (f) moving a sub-tour customer to the main tour and splitting the sub-tour.

search to the MC-TTRP. Therefore, this function contains both specific TTRP moves as well as the standard 2-opt, 2-opt*, exchange, and relocate operators commonly used in many VRP implementations. Figure 6 shows some of the moves performed in this step.

Finally, as shown in Algorithm 1, our heuristic for the MC-TTRP completes the work, returning a feasible route and the loading configuration of all the vehicles involved in the transport network.

4.3. Iterated tabu search

As stated before, in the second phase of our proposal, an iterated tabu search (ITS) is implemented. We developed this metaheuristic based on previous related works, such as that by Cordeau and Maischberger (2012) for some VRP variants and the approach implemented by Silvestrin and Ritt (2017) for the MC-VRP.

The ITS starts from a feasible solution, which, in our case, is the solution provided by the constructive heuristic. This procedure is usually based on two main actions: a tabu search and a perturbation. The former involves a set of local moves applied sequentially to improve the solution, with the ability to accept slightly worse solutions or to move through the infeasible region when the search is stuck. Moreover, to avoid returning to already-visited solutions, they use the short-term memory strategy known as the tabu list. In the case of the perturbation process, the objective is to escape from local optima by exploring the vicinity of the current solution through small changes in the routes. As a general overview, the ITS aims to improve the best-known solution by combining the diversification provided by the perturbation with the intensification and diversification produced by the tabu search.

Algorithm 2 describes the scheme of our proposed ITS, of which we will now present a brief outline. The input parameters are the solution returned by the CW, which is the starting point, and a maximum number of iterations used as a stopping criterion. In the main loop (lines 5–14), the current solution is perturbed at each iteration after having applied the tour improvement function (presented in Subsection 4.2). This modified solution will be used as an initial guess for the tabu

Algorithm 2. Iterated tabu search for the MC-TTRP

```
1: procedure ITS_MC-TTRP (cw_solution, max_iterITS)
2:    best_solution ← cw_solution
3:    current_solution ← cw_solution
4:    iterITS = 0
5:    while iterITS < max_iterITS do
6:       current_solution_i ← Improve tours in current_solution
7:       current_solution_i, ϕ ← Perturbs the current_solution_i
8:       current_solution_i, best_TS_solution ← TabuSearch(current_solution_i, ϕ, iterITS, max_iterITS)
9:       if cost(best_TS_solution) < cost(best_solution) then
10:         best_solution ← best_TS_solution
11:      end if
12:      with probability (iterITS/max_iterITS)^2, current_solution ← best_solution
13:      iterITS = iterITS + 1
14:   end while
15:   return(best_solution)
16: end procedure
```

search. In turn, as can be seen in line 8, this latter method returns its current solution, which will be the one to be perturbed in the next iteration, and its best solution, which can update the best overall solution (lines 9–11). In this way, throughout the iterations, the algorithm perturbs and searches the current solution, which is initially the best-known solution. However, as the algorithm progresses, there is a probability of working with another solution to avoid stagnation (line 12). Finally, the algorithm terminates when the stopping conditions are fulfilled, and it outputs the best solution found during the search.

Regarding the perturbation procedure, this is a key point in our proposal because it guarantees diversity in the method. First, a random number, ϕ, is chosen between 1% and 10% of the total number of customers. Then, ϕ customers are randomly selected and removed from the solution, to be subsequently reinserted. For the deletion and insertion operations, we use the generalized insertion procedure (GENI) and unstringing and stringing (US) algorithm proposed by Gendreau et al. (1992). This is followed by 3-opt and 4-opt* local search algorithms to improve the routes. Depending on the type of customer to be deleted or inserted, as well as its position in the solution, different possibilities of moves can arise. Some examples are shown in Fig. 7. Each customer is inserted into the route and position that minimizes the increase in the solution cost. If the perturbation failed to introduce the nodes that have been removed, we allow a threshold of infeasibility in the perturbed solution.

After the solution is perturbed, the algorithm performs the tabu search, whose implementation requires an additional explanation. This is an iterative procedure, where the strategy is essentially to apply the best possible MC-TTRP local search moves (some of which are shown in Fig. 6) in the neighborhood of the current solution, as long as these moves are not stored in the tabu list. The approach adopted for the tabu list is as follows. As it is known, a move involves a set of routes and the customers to be deleted or inserted into them. At each iteration, the tabu list stores for each move applied to the current solution, a set of pairs $\{custom i, rj\}$, with *customi* being the customer removed from route rj. In this way, the tabu search only accepts those moves where

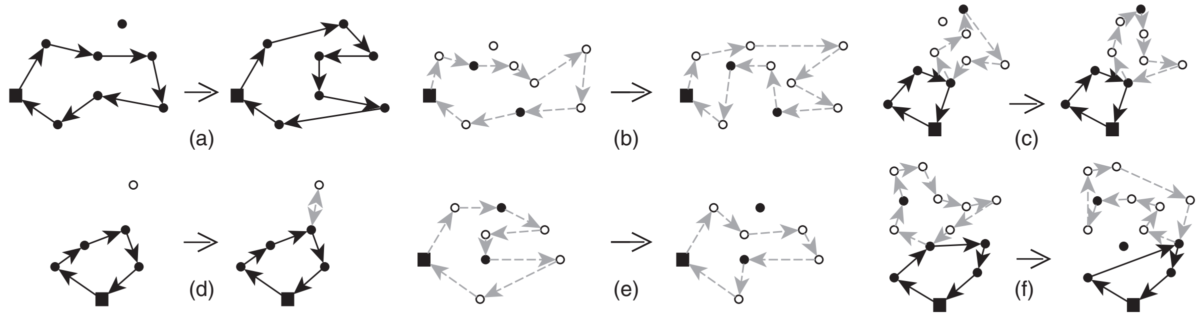


Fig. 7. Removal and insertion operations in the perturbation for the MC-TTRP. (a) Inserting a VC into a PVR, (b) inserting a TC into a PTR, (c) inserting a TC into a sub-tour, (d) converting a PVR into an MVR by creating a sub-tour with a single TC, (e) removing a VC from a PTR, and (f) removing a VC that works as parking place from an MVR.

all the customers introduced in the routes are different from all the pairs $\{customi, rj\}$ stored in the tabu list. The exception to this rule occurs when the result improves the best solution found so far, in which case an aspiration criterion is applied. Consequently, the tabu list prevents the insertion of customers into the same routes for a number τ of iterations. Each pair $\{customi, rj\}$ has an associated survival counter that controls the number of iterations that will be in the tabu list, and that requires to be updated at each iteration of the tabu search. When one of these counters reaches τ, its corresponding pair is removed from the tabu list. The parameter τ is initialized at the beginning of the tabu search and is chosen randomly from the uniform distribution on the interval $[1, \sqrt{n \cdot nr}]$, where n is the number of total customers, and nr corresponds to the number of routes in the current solution.

As the algorithm applies the best possible move, it may not improve the cost of the current solution or may even be infeasible. However, this is desirable because exploring new regions using worse or/and infeasible solutions prevents us from getting stuck in a local optimum. Therefore, to measure the total cost of each route, r, we consider the following objective function:

$$F(r) = d(r) + \alpha C(r), \tag{50}$$

where $d(r)$ is the total distance traveled, α a parameter of penalty, and $C(r)$ denotes the excess load on both vehicles and compartments. To calibrate the α penalty, we follow the approximation proposed by Cordeau and Maischberger (2012), which we recommend referring to for further details about this mechanism. Briefly, the α penalty is initially set to 1 and then updated throughout the tabu search, depending on the excess load in the current solution. In the case of feasibility, that is, if constraints are not violated, the penalty is decreased by a factor $1 + \gamma$. Otherwise, if the current solution has excess load on its routes, α is increased by $1 + \gamma$. The parameter γ is randomly selected at the beginning of the tabu search from the uniform distribution on [0,1]. Consequently, this update strategy produces an oscillatory effect between feasible and infeasible solutions.

In addition, as in other tabu search methods such as Cordeau and Maischberger (2012) or Silvestrin and Ritt (2017), a table with the most frequent moves is also used to penalize the function

Algorithm 3. Tabu search for the MC-TTRP

1: **procedure** TabuSearch (current_solution_i, ϕ, $iterITS$, $max\_iterITS$)
2: Choose τ randomly from U[1, $\sqrt{n \cdot nr}$] and create a new $tabuList = \{\}$
3: Create a new $freqPenList = \{\}$ and choose ζ randomly from U[0,1]
4: Initialize $\alpha = 1$ and choose γ randomly from U[0,1]
5: best_TS_solution ← current_solution_i
6: $counter\_iters\_without\_improvement = 0$
7: $max\_iters\_without\_improvement = \sqrt{(max\_iterITS - iterITS) \cdot \phi}$
8: **while** $counter\_iters\_without\_improvement < max\_iters\_without\_improvement$ **do**
9: Check all local search moves in the neighborhood of the current_solution_i
10: Evaluate the cost of the moves, penalizing infeasible solutions using α (see expression (50))
11: If the moves do not produce an improvement, they are penalized using ζ and $freqPenList$ (see expression (51))
12: Apply the best possible move in current_solution_i according to $tabuList$
13: **if** cost(current_solution_i) < cost(best_TS_solution) **then**
14: best_TS_solution ← current_solution_i
15: $counter\_iters\_without\_improvement = 0$
16: **else**
17: $counter\_iters\_without\_improvement = counter\_iters\_without\_improvement + 1$
18: **end if**
19: **if** current_solution_i is infeasible **then**
20: $\alpha = \alpha \cdot (1 + \gamma)$
21: **else**
22: $\alpha = \alpha/(1 + \gamma)$
23: **end if**
24: Save the selected move in $freqPenList$
25: Update $tabuList$
26: **end while**
27: **return**(current_solution_i, best_TS_solution)
28: **end procedure**

cost when the search is stuck in a local minimum. Thus, when the best possible move decreases the current solution cost, we reevaluate all possible moves according to a new objective function:

$$F'(r, M) = F(r)\left(1 + \zeta \sum_{ci \in cMr} freqPenList(ci, r)/i\right), \tag{51}$$

where M is a specific move, ζ is a penalty uniformly randomly chosen from [0,1], cMr is the set of customers involved in M to be inserted into a route r, $freqPenList$ is the table of frequencies, which returns how many times a customer ci entered in the route r, and i is the current iteration.

Algorithm 3 shows the main scheme for the implemented tabu search. In the first lines of the pseudocode (lines 2–7), the tabu list ($tabuList$) and the table with the most frequent moves ($freqPenList$) are created, and all the parameters explained above are initialized (τ, α, γ, and ζ). Furthermore, a maximum number of iterations without improvement is set (line 7), which, in this case, will be the stopping criterion. During the main loop (lines 8–26), the tabu search evaluates all possible moves, selecting the best possible one according to the constraints imposed by the tabu list. If the current solution is a local optimum, the algorithm selects the best move considering *freqPenList* and expression (51). Once a new solution is created, we check if it increases the cost of the best one found during the tabu search (lines 13–18). Next, α is calibrated based on the solution feasibility (lines 19–23). Moreover, the algorithm adds the selected move information to *freqPenList*. The tabu list is updated in line 25 by reducing the survival counter of the stored pairs. Notably, those pairs that reached the iteration limit inside the list are then released, and new pairs involved in the last applied move are included. Finally, after the algorithm reaches the stop condition, the tabu search returns both the best solution visited during the search and the current solution. The latter is from which Algorithm 2 will continue to work, perturbing it in the next iteration of the ITS algorithm.

5. Computational results and discussion

To evaluate the efficiency of our proposals, we analyzed the impact of the MC-TTRP model and two-phase heuristic using a modified version of a set of well-known benchmarks from the literature and a real-world problem.

The proposed heuristic described in Section 4 was implemented in `R` 4.0.2. To validate its performance, a set of experiments was conducted on the Finisterrae II supercomputer, provided by the Galicia Supercomputing Centre[4] (CESGA), which consists of 306 nodes powered by two deca-core Intel Haswell 2680v3 CPUs with 128 GB RAM connected through an Infiniband FDR network. The mathematical model presented in Subsection 3.2 was solved using the `Gurobi` 8.1.0 solver. The code was run on a hexa-core Intel i7-8700 CPU with 16 GB RAM.

The following subsections report the computational study. Subsection 5.1 reports a detailed study of the exact solution of the proposed MC-TTRP model using the case study. To the best of our knowledge, there are currently no existing MC-TTRP benchmark problems. Hence, before showing the computational results of our algorithm, Subsection 5.2 describes the generation of a set of new MC-TTRP test problems based on the 21 well-known TTRP cases introduced by Chao (2002). Subsection 5.3 describes the solutions obtained both in the instances created by Chao for the TTRP as well as in our generated datasets for the MC-TTRP using our proposed heuristic. Finally, the real-world application described in Subsection 3.1 was used to validate our heuristic, as reported in Subsection 5.4, comparing the quality of the results achieved to those in the case of the exact method.

5.1. Exact solving of a real example

As we have real data for this case study, an optimization scenario was built to assess our MC-TTRP model. In particular, we know the distances between the different nodes (customers and central depot) as well as customer demands and vehicle capacities. Moreover, the drivers work 8 hours per day, and we estimated the average vehicle speed to be 60 km/h. Furthermore, because we do not have information about the service time of each customer or the time employees need to load the trucks, we assumed it to be negligible. In addition, in our real instance of the model,

[4]https://www.cesga.es/

10 customers were selected, five of each type, provided by the company's vehicles: three trucks and two trailers.

The trucks have 13 hoppers each, which can carry up to 1.5 tons of loads, while the trailers have 15 hoppers each with a maximum capacity of 2 tons. The demand d of the customers (in kilograms), according to the four types of feed distributed by the cooperative, and the matrix of distances, C, (in kilometers) between the nodes are as follows:

$$d = \begin{pmatrix} 1000 & 0 & 0 & 2300 \\ 4000 & 0 & 0 & 2041 \\ 1959 & 0 & 4000 & 0 \\ 0 & 951 & 2000 & 0 \\ 0 & 3500 & 1385 & 0 \\ 0 & 3003 & 0 & 0 \\ 516 & 0 & 0 & 2500 \\ 978 & 0 & 3500 & 0 \\ 2000 & 0 & 2513 & 900 \\ 0 & 3490 & 0 & 0 \end{pmatrix} \text{ and } C = \begin{pmatrix} 0 & 21 & 20 & 17 & 65 & 63 & 60 & 19 & 22 & 24 & 60 \\ 21 & 0 & 4 & 6 & 60 & 58 & 55 & 15 & 18 & 20 & 55 \\ 20 & 4 & 0 & 4 & 59 & 56 & 53 & 13 & 8 & 12 & 53 \\ 17 & 6 & 4 & 0 & 57 & 54 & 52 & 11 & 13 & 16 & 52 \\ 65 & 60 & 59 & 57 & 0 & 3 & 7 & 66 & 69 & 71 & 6 \\ 63 & 58 & 56 & 54 & 3 & 0 & 4 & 64 & 66 & 69 & 3 \\ 60 & 55 & 53 & 52 & 7 & 4 & 0 & 61 & 64 & 66 & 2 \\ 19 & 15 & 13 & 11 & 66 & 64 & 61 & 0 & 3 & 5 & 61 \\ 22 & 18 & 8 & 13 & 69 & 66 & 64 & 3 & 0 & 7 & 64 \\ 24 & 20 & 12 & 16 & 71 & 69 & 66 & 5 & 7 & 0 & 66 \\ 60 & 55 & 53 & 52 & 6 & 3 & 2 & 61 & 64 & 66 & 0 \end{pmatrix}.$$

To solve the associated mathematical problem, `Gurobi`[5] was used as an exact MILP solver. The optimization process required 31.2 hours to obtain the optimal solution. The value of the objective function was 207 km, containing an MVR of 74 km, whose main route, 0-3-2-1-0, is traveled by truck 1 with trailer 1 attached. Customer 2 serves as a trailer parking place for the sub-tour 2-8-7-9-2 with the truck. The remaining customers are served by truck 3 on a PTR, 0-6-5-4-10-0.

It can be seen that only a trailer is required to supply these 10 customers. We studied the effect of trailers in the solution and compared the results with those obtained if only trucks were used. In such a case, and after a runtime of 6.23 hours, we obtained the following results: the value of the objective function was 232 km; truck 1 distributes feed to customers 1–3, traveling 46 km; truck 2 travels 53 km and serves customers 7–9; and truck 3 performs the route 0-4-5-10-6-0, whose length is of 133 km.

As can be seen, when the company introduces trailers, a reduction is achieved not only in the total length traveled (which decreases by 25 km) but also in the number of drivers required (which decreases from 3 to 2).

Moreover, we can deduce that when using only trucks, we are faced with an MC-VRP. However, because of the large amount of feed that the cooperative must distribute daily, we suggest the incorporation of trailers to accommodate the use of the MC-TTRP model. Furthermore, given the existence of access restrictions to some farms, this model seems appropriate. The purchase or rental of these additional vehicles can indeed be a significant initial investment, but the benefits usually compensate in the long term because, among other things, it is not necessary to hire more drivers.

To study the increase in the computation time when the instance is slightly modified, we tried to solve this problem when adding a new VC. By having one more node, the number of variables involved increases considerably, and this causes the execution time to exceed two weeks.

[5]https://www.gurobi.com/

Agricultural cooperatives often must supply a large number of members, and it is not feasible to take more than 15 days to solve a problem with 11 customers. As stated in Subsection 3.1, the cooperative that motivated our study receives orders from approximately 40 customers per day. Furthermore, certain occurrences could require a sudden reorganization of the routes. This issue encouraged us to consider designing the heuristic described above, which quickly and efficiently solves the problem of multi-compartmental trucks and trailers.

5.2. Set of instances

To the best of our knowledge, there are no benchmark instances in the literature for the MC-TTRP. Therefore, it was necessary to slightly modify the 21 TTRPs[6] developed by Chao (2002). Thus, to test our algorithm, we generated 21 new MC-TTRP instances by adding compartments to Chao's benchmarks. These test problems were derived from seven VRPs created by Christofides et al. (1979), with 50 to 199 customers by specifying 25%, 50%, and 75% of the customers as truck customers. For further details about the nature of these problems, we recommend referring to the original work of Chao (2002).

Regarding the characteristics of our MC-TTRPs, Table 2 gives a general overview of our instances[7], which present variety in terms of the number of customers and vehicles, vehicle capacity, and compartment configuration. The strategy chosen to generate them was as follows: first, the number of available vehicles is computed as $\lceil 1.5 \cdot av \rceil$, where av denotes the corresponding available vehicle in Chao's TTRPs. Also, the original capacity of each truck and trailer was divided into compartments of capacity 5 and 10, respectively. As a result, depending on the instance, trucks could be split into 20 or 30 compartments, while trailers were all split into 10. Furthermore, we assumed that every customer demands two different types of products, so we equally split the total original demand.

Moreover, we classified these instances into three categories depending on their percentage of truck customers:

- G1: instances 1, 4, 7, 10, 13, 16, and 19 belong to group one.
- G2: instances 2, 5, 8, 11, 14, 17, and 20 belong to group two.
- G3: instances 3, 6, 9, 12, 15, 18, and 21 belong to group three.

5.3. Results on test instances

The following computational results help to evaluate the performance of the proposed heuristic. Having shown in Subsection 5.1 that the exact solving of a medium-sized problem can be extremely computation-intensive, we highlight that the main focus of this study was to obtain good-quality solutions in reasonable computation time. Moreover, to the best of our knowledge, there are no other solution methods for MC-TTRP in the literature.

[6]The 21 test problems of the TTRP (Chao, 2002) are available at http://web.ntust.edu.tw/~vincent/ttrp/.

[7]Note that every three consecutive instances come from the same VRP.

Table 2
Dimensions of the MC-TTRP test problems

Problem number	Customers VCs	Customers TCs	Trucks Number	Trucks Compart.	Trucks Capacity of compartments	Trucks Total capacity	Trailers Number	Trailers Compart.	Trailers Capacity of compartments	Trailers Total capacity
1	38	12								
2	25	25	8	20	5	100	5	10	10	100
3	13	37								
4	57	18								
5	38	37	14	20	5	100	8	10	10	100
6	19	56								
7	75	25								
8	50	50	12	30	5	150	6	10	10	100
9	25	75								
10	113	37								
11	75	75	18	30	5	150	9	10	10	100
12	38	112								
13	150	49								
14	100	99	26	30	5	150	14	10	10	100
15	50	149								
16	90	30								
17	60	60	11	30	5	150	6	10	10	100
18	30	90								
19	75	25								
20	50	50	15	30	5	150	8	10	10	100
21	25	75								

Therefore, to obtain an initial validation of the quality of our CW and ITS, we analyzed their performance with the TTRP instances described above in Table 2 but without using compartments. Notably, in this manner, it is possible to compare the solutions obtained by our proposed method with those reported by Chao (2002) and Caramia and Guerriero (2010a).

Table 3 summarizes the results obtained for both phases of the algorithms, where the first column represents the instance number, and columns 2, 3 and 4 show the objective value returned by the constructive phase of the three approaches considered, that is, the distance covered by all routes once an initial feasible solution is achieved. Note that we have considered the objectives reported by Chao when a descent improvement subroutine is implemented. Concerning the solutions obtained by the constructive stage, our savings-based heuristic allowed the creation of feasible routes, improving on the objectives reported by Chao and Caramia and Guerriero in some cases, and outperforming both of them in five problems (bold values of column 2).

Furthermore, the next four columns indicate the objective achieved in the improvement phase, where BKS denotes the best known solution values according to Caramia and Guerriero (2010a). For Chao and Caramia and Guerriero's algorithms, values represent their best obtained solutions, with no further details given. In our case, we carried out 500 iterations of the ITS and report the best value over 10 runs. Bolds values of column 5 indicate those problems in which our ITS

Table 3
Computational results for 21 test TTRPs

Problem number	Constructive phase			Improvement phase				Number of vehicles	
	Our CW	Chao	Caramia	Our ITS	Chao	Caramia	BKS	Trucks	Trailers
1	**632.38**	646.02	645.72	**565.01**	565.02	566.80	564.68	5	3
2	711.93	739.90	699.68	635.28	658.07	620.15	612.75	5	3
3	805.34	774.78	770.19	634.16	648.74	632.48	618.04	5	3
4	1002.58	943.47	902.89	809.29	856.20	803.32	798.53	9	5
5	1067.48	1130.85	1035.89	844.46	949.98	842.50	839.62	9	5
6	1193.98	1236.69	1171.50	981.08	1053.23	938.18	933.26	9	5
7	**899.74**	906.31	902.18	852.93	832.26	832.56	830.48	8	4
8	1028.96	971.60	970.45	887.26	900.54	878.87	878.36	8	4
9	1138.93	1106.66	1082.99	**958.50**	971.62	980.42	934.47	8	4
10	1213.15	1159.78	1151.23	1079.73	1073.50	1060.41	1039.07	12	6
11	**1276.23**	1288.74	1288.74	**1123.43**	1170.17	1170.70	1094.11	12	6
12	**1408.93**	1453.82	1440.12	1224.71	1217.01	1178.34	1155.13	12	6
13	1530.66	1481.40	1480.46	1344.82	1364.50	1288.46	1287.18	17	9
14	1617.71	1624.96	1612.34	1444.89	1464.20	1372.52	1353.08	17	9
15	**1706.19**	1858.87	1752.94	1499.29	1540.25	1470.21	1457.61	17	9
16	1349.67	1267.87	1060.26	1015.26	1041.36	1004.69	1002.49	7	4
17	1376.31	1261.17	1120.34	1073.77	1090.46	1042.35	1042.35	7	4
18	1439.34	1366.21	1220.25	1205.46	1141.36	1129.16	1129.16	7	4
19	1141.59	969.96	880.22	852.01	854.02	813.50	813.50	10	5
20	1097.47	1140.47	961.25	908.87	942.39	848.93	848.93	10	5
21	1085.61	1174.43	1009.68	930.70	926.47	909.06	909.06	10	5

outperformed the other two approaches. Finally, the last two columns indicate the corresponding available numbers of trucks and trailers.

With respect to computation times, our algorithm requires from 7 to 200 min to achieve the best solutions reported in Table 3. Note that our heuristic is efficient to solve the TTRP, but it is actually designed to solve the MC-TTRP. We present the above comparison with existing TTRP solutions in an indicative way to show that our algorithm performs well on problems that may be similar.

Returning to the specific problem of interest, Table 4 shows the performance of the proposed heuristic, on the 21 instances of the MC-TTRP described in Table 2, considering the hoppers properly. The first columns indicate the problem number and the objective of the constructive phase. The remaining columns refer to the ITS solution. Columns 3–8 show the objective of the improvement phase (after 100 iterations of the ITS, reporting the best value over 10 runs), vehicles used, and different routes created. The next columns list the vehicle occupancy rate. Column 9 shows the number of truck compartments used divided by the total number of truck compartments in the solution. For instance, the total number of truck compartments in problem 1 was 160 (eight trucks used with 20 compartments each). Column 10 is analogous, considering trailers instead of trucks. The "Total" column is the quotient of the sum of used truck and trailer compartments divided by the total number of compartments in the solution. That is, for problem 1, the total number is

Table 4
Computational results for 21 test MC-TTRPs

Problem number	CW objective	ITS objective	No. of used vehicles		Number of routes			Occupancy rate			Time (min)
			Trucks	Trailers	PTR	PVR	MVR	Trucks	Trailers	Total	
1	662.93	640.66	8	5	3	4	1	0.79	0.84	0.80	5.58
2	832.60	708.08	8	4	4	0	4	0.88	0.85	0.88	5.28
3	885.79	764.81	8	3	5	0	3	1	0.73	0.96	5.76
4	1013.57	908.53	13	8	5	5	3	0.85	0.88	0.85	10.55
5	1168.01	1012.83	14	7	7	3	4	0.85	0.83	0.84	8.93
6	1264.09	1101.90	14	7	7	1	6	0.92	0.69	0.87	14.13
7	1013.72	938.90	11	6	5	4	2	0.89	0.90	0.89	28.8
8	1065.39	1003.95	12	5	7	4	1	0.84	0.84	0.84	24.59
9	1357.49	1113.95	11	5	6	0	5	0.96	0.80	0.94	23.82
10	1336.30	1275.87	18	9	9	7	2	0.83	0.80	0.83	49.13
11	1422.92	1367.46	18	9	9	4	5	0.87	0.73	0.85	46.72
12	1683.38	1434.51	18	6	12	0	6	0.92	0.70	0.90	62.71
13	1670.52	1584.96	23	14	9	10	4	0.87	0.89	0.87	93.38
14	1858.73	1836.16	26	14	12	4	10	0.85	0.64	0.82	90.90
15	2072.07	1958.46	25	11	14	0	11	0.90	0.75	0.88	83.76
16	1681.62	1471.36	11	6	5	2	4	0.89	0.87	0.89	41.73
17	1891.04	1606.92	11	5	6	1	4	0.95	0.84	0.94	45.00
18	1968.90	1590.13	11	4	7	0	4	0.96	1	0.97	60.41
19	1026.89	885.93	12	8	4	3	5	0.70	0.90	0.74	22.69
20	1125.27	1037.27	13	7	6	3	4	0.70	0.83	0.72	17.24
21	1115.85	978.19	12	3	9	0	3	0.89	0.80	0.89	24.31

210 (160 truck plus 50 trailer compartments). Finally, the last column refers to the running time in minutes.

To complement the analysis presented in Table 4, it should be noted that the occupancy rates are relatively high, with averages of 87%, 81%, and 87% for trucks, trailers, and total, respectively. To provide a better interpretation of the results, we calculated the average rates for groups G1, G2, and G3. The values obtained for trucks were 83%, 85%, and 94%, respectively; thus, we can see that there is a growth in the truck occupancy rate as the number of truck customers increases. The opposite applies in the case of trailers: the rates are 87%, 79%, and 78% for groups G1–G3, respectively. These results are promising because feasible solutions were achieved at low computational cost.

5.4. Application to real-world data

To evaluate the quality of our heuristic algorithm, we considered the data of the real-world example described in Subsection 3.1. We solved a set of different-sized problems by combining customers of the agricultural cooperative: P1–P3 have four customers of each class; P4 and P5 have nine customers, five of which are VCs; and P6–P8 involve 10 customers, five VCs and five TCs. All of

Table 5
Results obtained by the exact model versus results obtained with our heuristic algorithm (after 100 iterations of the ITS)

Problem number	Size	Method	Objective	Occupancy rate			Time (s)
				Trucks	Trailers	Total	
P1	8	Exact	189	0.92	0.80	0.88	309.27
		Heuristic	189	0.88	0.87	0.88	5.03
P2	8	Exact	140	0.81	0.73	0.78	6733.20
		Heuristic	142	0.81	0.73	0.78	3.88
P3	8	Exact	106	0.77	0.67	0.73	1275.75
		Heuristic	106	0.77	0.67	0.73	3.90
P4	9	Exact	256	0.85	0.80	0.83	16,766.7
		Heuristic	256	0.81	0.87	0.83	5.58
P5	9	Exact	109	0.81	0.80	0.80	17,511
		Heuristic	109	0.62	1	0.76	4.72
P6	10	Exact	222	0.96	0.67	0.85	221,019
		Heuristic	223	0.92	0.67	0.83	6.48
P7	10	Exact	237	0.92	0.80	0.88	671,316
		Heuristic	238	0.88	0.87	0.88	7.97
P8	10	Exact	207	0.96	0.67	0.85	112,389
		Heuristic	213	0.69	1	0.80	6.09

Table 6
Lower and upper bounds and gaps for the exact formulation

Problem number	Lower bound	Upper bound	Relative gap
P1	189	189	0
P2	140	140	0
P3	106	106	0
P4	256	256	0
P5	109	109	0
P6	198	222	0.11
P7	185	237	0.22
P8	73	207	0.65

these instances imply two trucks with 13 hoppers that can contain up to 1.5 tons each and one trailer with 15 hoppers with a maximum capacity of 2 tons. Given this information, Table 5 gives a comparison between the solutions of both methods. Furthermore, Table 6 provides the lower and upper bounds and relative gaps for the exact formulations of those 8 problems by setting a time limit of three hours. It can be seen that problems with 10 customers present a positive gap, which in the particular case of P8 is relatively large.

Table 5 shows that our algorithm achieved the exact objective for problems P1, P3, and P4, although the loading procedures were not always the same. Our heuristic did not achieve the exact optimal values for the remaining instances, but there was an increase of at most 2.90% in the objective (which occurred for P8). However, the execution times of the heuristic were significantly shorter (by several orders of magnitude) than those of the exact method. Although optimality convergence

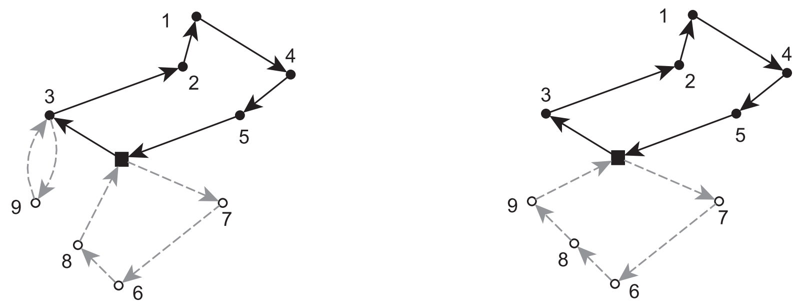


Fig. 8. Solutions obtained by `Gurobi` (left) and the two-phase heuristic (right) for problem P5.

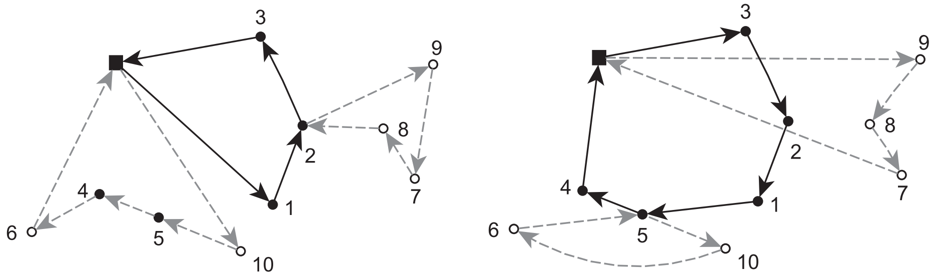


Fig. 9. Solutions obtained by `Gurobi` (left) and the two-phase heuristic (right) for problem P8.

was not achieved for all problems, these results are promising because a feasible solution (close to the optimal one) was obtained at a much smaller computational cost (in less than 10 seconds).

Moreover, the number of vehicles and types of routes were the same for both methods in every problem, except for P5; in this case, our approach served all VCs in an MVR, while TCs were served by a PTR. However, the exact solution allocated one of the TCs to a sub-tour with customer 3 as the trailer parking place. Figure 8 illustrates the results for this instance.

As another illustrative example of the difference between the exact solution and the heuristic, we show the route configuration for P8 in Fig. 9. The customers of the sub-tour presented in the optimal solution, 2–9–7–8–2, are part of a PTR in our algorithm's result. Customer 5 was selected the trailer parking place for the sub-tour of the heuristic solution, 5–10–6–5.

Table 5 also indicates that, as opposed to the exact result, the heuristic approach gives preference to the loading of trailers, having an average occupancy rate of 86%, while the trucks are 78% full on average. The total occupancy rate does not differ significantly between these two methods.

6. Conclusions

This study considered a routing problem for trucks and trailers divided into compartments, called the MC-TTRP. We proposed a novel mathematical formulation of this model as an MILP problem and presented a detailed explanation of its constraints. A real-world example of a Spanish agricultural cooperative that distributes feed to its customers was considered to verify the proposed model. Given that exactly solving the problem for large-sized instances is computationally expensive, the use of operational research techniques is necessary to obtain solutions. Therefore, we introduced and implemented a new heuristic algorithm to reduce computation time. Our proposal is a two-stage approach: the first phase iteratively builds an initial solution, based on the savings method of Clarke and Wright, and then the second phase aims to refine the solution. To achieve this, an iterative tabu search was designed. We conducted a thorough computational study on different instances. First, the 21 benchmark problems of Chao (2002) were analyzed and the solutions reported by both Chao (2002) and Caramia and Guerriero (2010a) were compared to ours. In addition, we suitably adapted these data sets to consider compartments, leading to 21 challenging test problems. Our heuristic always generated a feasible solution to the test problems, and the results obtained showed that our method can effectively and efficiently solve the MC-TTRP. Furthermore, this algorithm was applied to the previously mentioned real-world case of a cooperative. A comparison between our results and those provided by the exact formulation showed a significant decrease in computational cost, achieving good-quality solutions to the problems. Considering this, we believe that the proposed heuristic is a promising solution approach for the MC-TTRP.

There is still much room for further research on this variant of the VRP. It would be interesting to apply the model and solution algorithm to other feed producing companies similar to that considered in this study as well as other businesses, such as milk collection and fuel distribution companies. Another open research direction is to extend the model to include modifications, such as stochastic demands, time windows, or heterogeneous vehicle fleets, to address other similar real-world problems. In addition, future work could be to develop other metaheuristic methods, such as simulated annealing or evolutionary algorithms, that take our solution as a benchmark to compare with it. Finally, in view of the different ingredients that make up our model, including route design, assignment of customers to vehicles, and loading of compartments, exploring the performance of decomposition techniques for solving it could provide good results.

The code and instances required to reproduce the results reported herein are available at https://github.com/LauraDavilaPena/ITS_MC-TTRP

Acknowledgments

We acknowledge the computational resources provided by CESGA. Laura Davila-Pena's research was funded by the Ministry of Education, Culture and Sports of Spain (contract FPU17/02126). David R. Penas' research was funded by the *Xunta de Galicia* (post-doctoral contract ED481B-2019-010). This work was also supported by the ERDF (MINECO/AEI grant MTM2017-87197-C3-3-P) and by the *Xunta de Galicia* (Competitive Reference Group ED431C 2017/38 and ED431C 2021/24). We would also like to thank the three anonymous referees for their constructive comments and suggestions, which helped us to improve the final version of this paper.

References

Accorsi, L., Vigo, D., 2020. A hybrid metaheuristic for single truck and trailer routing problems. *Transportation Science* 54, 5, 1351–1371.

Alinaghian, M., Shokouhi, N., 2018. Multi-depot multi-compartment vehicle routing problem, solved by a hybrid adaptive large neighborhood search. *Omega* 76, 85–99.

Alvarez, A., Munari, P., Morabito, R., 2018. Iterated local search and simulated annealing algorithms for the inventory routing problem. *International Transactions in Operational Research* 25, 6, 1785–1809.

Brown, G.G., Ellis, C.J., Graves, G.W., Ronen, D., 1987. Real-time, wide area dispatch of mobil tank trucks. *Interfaces* 17, 1, 107–120.

Brown, G.G., Graves, G.W., 1981. Real-time dispatch of petroleum tank trucks. *Management Science* 27, 1, 19–32.

Caramia, M., Guerriero, F., 2010a. A heuristic approach for the truck and trailer routing problem. *Journal of the Operational Research Society* 61, 7, 1168–1180.

Caramia, M., Guerriero, F., 2010b. A milk collection problem with incompatibility constraints. *Interfaces* 40, 2, 130–143.

Chao, I.M., 2002. A tabu search method for the truck and trailer routing problem. *Computers & Operations Research* 29, 1, 33–51.

Chao, I.M., Golden, B.L., Wasil, E.A., 1998. A new algorithm for the site-dependent vehicle routing problem. In Woodruff, D. (ed.), *Advances in Computational and Stochastic Optimization, Logic Programming, and Heuristics Search: Interfaces in Computer Science and Operations Research*. Springer, Boston, pp. 301–312.

Christofides, N., Mingozzi, A., Toth, P., 1979. The vehicle routing problem. In Christofides, N., Mingozzi, A., Toth, P. and Sandi, C. (eds), *Combinatorial Optimization*. Wiley, Chichester, pp. 315–338.

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

Coelho, L.C., Laporte, G., 2015. Classification, models and exact algorithms for multi-compartment delivery problems. *European Journal of Operational Research* 242, 3, 854–864.

Cordeau, J.F., Maischberger, M., 2012. A parallel iterated tabu search heuristic for vehicle routing problems. *Computers & Operations Research* 39, 9, 2033–2050.

Davila-Pena, L., 2019. Modelos y algoritmos en una clase de problemas de rutas de vehículos. Master's thesis, University of Santiago de Compostela.

Derigs, U., Gottlieb, J., Kalkoff, J., Piesche, M., Rothlauf, F., Vogel, U., 2011. Vehicle routing with compartments: applications, modelling and heuristics. *OR Spectrum* 33, 4, 885–914.

Derigs, U., Pullmann, M., Vogel, U., 2013. Truck and trailer routing problems, heuristics and computational experience. *Computers & Operations Research* 40, 2, 536–546.

El Fallahi, A., Prins, C., Calvo, R.W., 2008. A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. *Computers & Operations Research* 35, 5, 1725–1741.

Gendreau, M., Hertz, A., Laporte, G., 1992. New insertion and postoptimization procedures for the traveling salesman problem. *Operations Research* 40, 6, 1086–1094.

Gerdessen, J.C., 1996. Vehicle routing problem with trailers. *European Journal of Operational Research* 93, 1, 135–147.

Glover, F., Laguna, M., 1998. Tabu search. In Pardalos, P. (ed.), *Handbook of Combinatorial Optimization*. Springer, Boston, pp. 2093–2229.

Guitián de Frutos, R.M., Casas-Méndez, B., 2019. Routing problems in agricultural cooperatives: a model for optimization of transport vehicle logistics. *IMA Journal of Management Mathematics* 30, 4, 387–412.

Henke, T., Speranza, M.G., Wäscher, G., 2015. The multi-compartment vehicle routing problem with flexible compartment sizes. *European Journal of Operational Research* 246, 3, 730–743.

Henke, T., Speranza, M.G., Wäscher, G., 2019. A branch-and-cut algorithm for the multicompartment vehicle routing problem with flexible compartment sizes. *Annals of Operations Research* 275, 2, 321–338.

Lespay, H., Suchan, K., 2021. A case study of consistent vehicle routing problem with time windows. *International Transactions in Operational Research* 28, 3, 1135–1163.

Lin, S.W., Vincent, F.Y., Lu, C.C., 2011. A simulated annealing heuristic for the truck and trailer routing problem with time windows. *Expert Systems with Applications* 38, 12, 15244–15252.

Lin, S.W., Yu, V.F.Y., Chou, S.Y., 2009. Solving the truck and trailer routing problem based on a simulated annealing heuristic. *Computers & Operations Research* 36, 5, 1683–1692.

Mendoza, J.E., Castanier, B., Guéret, C., Medaglia, A.L., Velasco, N., 2010. A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands. *Computers & Operations Research* 37, 11, 1886–1898.

Mendoza, J.E., Castanier, B., Guéret, C., Medaglia, A.L., Velasco, N., 2011. Constructive heuristics for the multicompartment vehicle routing problem with stochastic demands. *Transportation Science* 45, 3, 346–363.

Muyldermans, L., Pang, G., 2010. On the benefits of co-collection: Experiments with a multi-compartment vehicle routing algorithm. *European Journal of Operational Research* 206, 1, 93–103.

Ostermeier, M., Henke, T., Hübner, A., Wäscher, G., 2021. Multi-compartment vehicle routing problems: State-of-the-art, modeling framework and future directions. *European Journal of Operational Research* 292, 3, 799–817.

Ostermeier, M., Hübner, A., 2018. Vehicle selection for a multi-compartment vehicle routing problem. *European Journal of Operational Research* 269, 2, 682–694.

Parragh, S.N., Cordeau, J.F., 2017. Branch-and-price and adaptive large neighborhood search for the truck and trailer routing problem with time windows. *Computers & Operations Research* 83, 28–44.

Renaud, J., Boctor, F.F., Laporte, G., 1996. A fast composite heuristic for the symmetric traveling salesman problem. *INFORMS Journal on Computing* 8, 2, 134–143.

Rothenbächer, A.K., Drexl, M., Irnich, S., 2018. Branch-and-price-and-cut for the truck-and-trailer routing problem with time windows. *Transportation Science* 52, 5, 1174–1190.

Scheuerer, S., 2004. Neue Tabusuche-Heuristiken für die logistische Tourenplanung bei restringierendem Anhängereinsatz, mehreren Depots und Planungsperioden. PhD thesis, University of Regensburg.

Scheuerer, S., 2006. A tabu search heuristic for the truck and trailer routing problem. *Computers & Operations Research* 33, 4, 894–909.

Semet, F., 1995. A two-phase algorithm for the partial accessibility constrained vehicle routing problem. *Annals of Operations Research* 61, 1, 45–65.

Semet, F., Taillard, E., 1993. Solving real-life vehicle routing problems efficiently using tabu search. *Annals of Operations Research* 41, 4, 469–488.

Silvestrin, P.V., Ritt, M., 2017. An iterated tabu search for the multi-compartment vehicle routing problem. *Computers & Operations Research* 81, 192–202.