Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 6012105, 19 pages
https://doi.org/10.1155/2022/6012105

Research Article

Mathematical Model and Simulated Annealing Algorithm for the Two-Dimensional Loading Heterogeneous Fixed Fleet Vehicle Routing Problem

Lilian Caroline Xavier Candido [1,2] **and Luzia Vidal de Souza** [1]

[1] *Federal University of Parana, Curitiba, Brazil*
[2] *Federal University of Technology – Parana, Campo Mourão, Brazil*

Correspondence should be addressed to Lilian Caroline Xavier Candido; liliancarolinex@gmail.com

Received 17 June 2021; Accepted 17 December 2021; Published 28 January 2022

Academic Editor: Arunava Majumder

This paper addresses the two-dimensional loading heterogeneous fixed fleet vehicle routing problem, which is a complex and unstudied variant of the classical vehicle routing problem and has a wide range of applications in transportation and logistics fields. In this problem, each customer demands a set of rectangular two-dimensional items, and the objective is to find the minimum cost delivery routes for a limited set of vehicles with different capacities, fixed and variable operating costs, and rectangular two-dimensional loading surfaces. We formulate a mixed integer linear programming model to obtain optimal solutions for small-scale problems. To obtain solutions for large-scale problems, we develop an algorithm based on simulated annealing and local search, which uses a collection of packing heuristics to address the loading constraints, and we also propose three new heuristics. We conduct experiments on benchmark instances derived from the two-dimensional loading heterogeneous fleet vehicle routing problem. The results indicate that the proposed model correctly describes the problem and can solve small-scale problems, that the new packing heuristics are effective in improving the collection of packing heuristics, and that the proposed simulated annealing algorithm can find good solutions to large-scale problems within an acceptable computational time. Hence, it can be used by logistic companies using a heterogeneous fixed fleet in the integrated planning of vehicle loading and routing.

1. Introduction

In many organizations, the management of distribution activities is a major decision-making problem. Every manufacturing system needs an efficient way to supply its products to retailers and customers. Most firms require delivery vehicles to service a network of demand locations, meaning the efficient use of a vehicle fleet is the main feature of almost all distribution problems, since transportation costs and lead time have an important impact on supply chain management [1, 2].

The vehicle routing problem (VRP) is considered one of the most important factors in both logistics and freight transportation systems. It consists of proposing the most cost-effective way to deliver items from depots to customers using a fleet of vehicles. The most common costs associated with the problem are related to driving distance. There are several variants of the VRP due to additional constraints encountered in real-world applications [3]. Regarding fleet composition, the classical problem, called the capacitated vehicle routing problem (CVRP), considers vehicles with the same limited capacity, whereas the heterogeneous fleet vehicle routing problem (HFVRP) considers vehicles with different capacities and costs [4]. Different features of these heterogeneous VRPs have been studied. For instance, the fleet size and mix vehicle routing problem (FSMVRP) deals with an unlimited number of vehicles, and in this case, in addition to routing, the problem includes the management of the fleet composition, while the heterogeneous fixed fleet vehicle routing problem (HFFVRP) considers a limited number of vehicles.

In the above VRP variants, customer demands are usually expressed simply as weights or volumes. In this case, checking the feasibility of a solution simply requires one to ensure that the sum of the customer demands assigned to each vehicle does not exceed its total loading capacity. However, a variety of real-life applications in the distribution management context involve the transportation of rectangular-shaped items, that is, items that cannot be stacked due to their weight or fragility, such as household appliances or delicate pieces of furniture. In all these cases, it is necessary to consider additional constraints to reflect the two-dimensional loading feature of the problem, because the way these items are packed into vehicles might have a significant influence over distribution costs. When dealing with rigid discrete items, their geometry may lead to infeasible solutions if the vehicle does not have sufficient capacity. In other words, it may not be possible to accommodate items in vehicles because of their geometrical characteristics, so logistic managers must simultaneously deal with routing and packing [5–7]. Since routing and packing are well known NP-hard problems, combining them leads to an extremely challenging optimization problem.

The two-dimensional loading capacitated vehicle routing problem (2L-CVRP) is one of the first approaches to integrate vehicle routing and loading problems. In this problem, customers demand a set of rectangular two-dimensional items, and identical vehicles have a two-dimensional loading surface. In real-world problems, in contrast to the 2L-CVRP, enterprises own a diverse fleet of vehicles, which offers the flexibility to design a more profitable distribution plan. The two-dimensional loading heterogeneous fleet vehicle routing problem (2L-HFVRP) treats the 2L-CVRP with an unlimited heterogeneous fleet. Nevertheless, since most companies that have to deliver goods own a limited fleet of vehicles, it is crucial to study routing problems that involve heterogeneous fixed fleets and loading constraints. To the best of our knowledge, we are not aware of studies that have been conducted to address such a VRP, although it is a practical problem in real-world transportation and logistics enterprises.

In this paper, we thus combine the HFFVRP with two-dimensional loading constraints, which is called the two-dimensional loading heterogeneous fixed fleet vehicle routing problem (2L-HFFVRP). In the 2L-HFFVRP, there are limited numbers of vehicles of different types. Each vehicle has a carrying capacity, a fixed cost related to the use of the vehicle, a variable cost proportional to the distance traveled, and a rectangular loading surface with a given length and width. Customer demand is defined by a set of rectangular items of a given width, length, and weight. All items belonging to a particular customer must be assigned to the same route. The objective is to describe the most cost-effective item delivery routes that start from and return to a depot. In practice, the need for different types of vehicles is determined by customer characteristics. Usually, larger vehicles are more appropriate for serving customers who require large orders, while smaller vehicles are more adequate for delivering small quantities or serving customers that have access restrictions.

Due to their structure, it happens frequently that the items may not be picked up from any side by the loading/unloading equipment. Additionally, vehicles are generally rear-loaded, and load rearrangement at the customer site can be difficult, time-consuming, or even impossible due to the weight and size of the items or the limitations of forklift trucks; therefore, each item to be unloaded must not be blocked by other items yet to be unloaded, even partially, in the rectangular area from its loading position to the unloading side of the truck. From the viewpoint of the loading problem, there are different constraints that could be considered. Depending on these constraints, it is possible to distinguish between the following loading settings: (a) oriented loading (OL), where the rotation of items is not allowed, that is, it is assumed that all items have a fixed orientation; (b) nonoriented loading (RL), where it is allowed to rotate items by 90° during the packing process; (c) sequential loading (SL), where items are always loaded in a reverse order to the order in which customers are visited but rearrangements of the items inside the vehicle are not allowed once the route has started; and (d) unrestricted loading (UL), where items are allowed to be rearranged during the distribution process. Figure 1 illustrates the difference between unrestricted and sequential loading.

The purpose of this paper is to present two approaches for solving 2L-HFFVRP. First, a mixed integer linear programming model is formulated to obtain optimal solutions for small-scale problems. To this end, we developed a four-index formulation that simultaneously manages the routing and loading aspects and also considers the sequential loading constraints. Second, to obtain solutions for large-scale problems, we propose an algorithm based on simulated annealing (SA) and local search. Specifically, we developed a heuristic algorithm to obtain an initial feasible solution and used SA and local search to explore the routing problem search space, while the loading constraints were solved by a bundle of packing heuristics composed of five heuristics from the literature and three new heuristics. We assume that all items have a fixed orientation and consider both unrestricted and sequential loading.

The remainder of this paper is organized as follows. Section 2 briefly reviews the literature on combined two-dimensional loading and routing problems. In Section 3, we define the problem to be addressed. In Section 4, we present the mathematical formulation for this problem. In Section 5, the packing heuristics are described, and a hybrid SA algorithm is developed. Section 6 presents the set of instances generated and explains the computational experiments and results. Finally, Section 7 provides the managerial insights of the study and Section 8 draws concluding remarks and perspectives for future research.

2. Literature Review

Integrated vehicle routing and loading problems have been increasingly studied, owing to their relevance to logistics distribution systems [5–7]. To the best of our knowledge, there are no studies on the 2L-HFFVRP to date. This section will focus on the previous studies on the 2L-CVRP and the

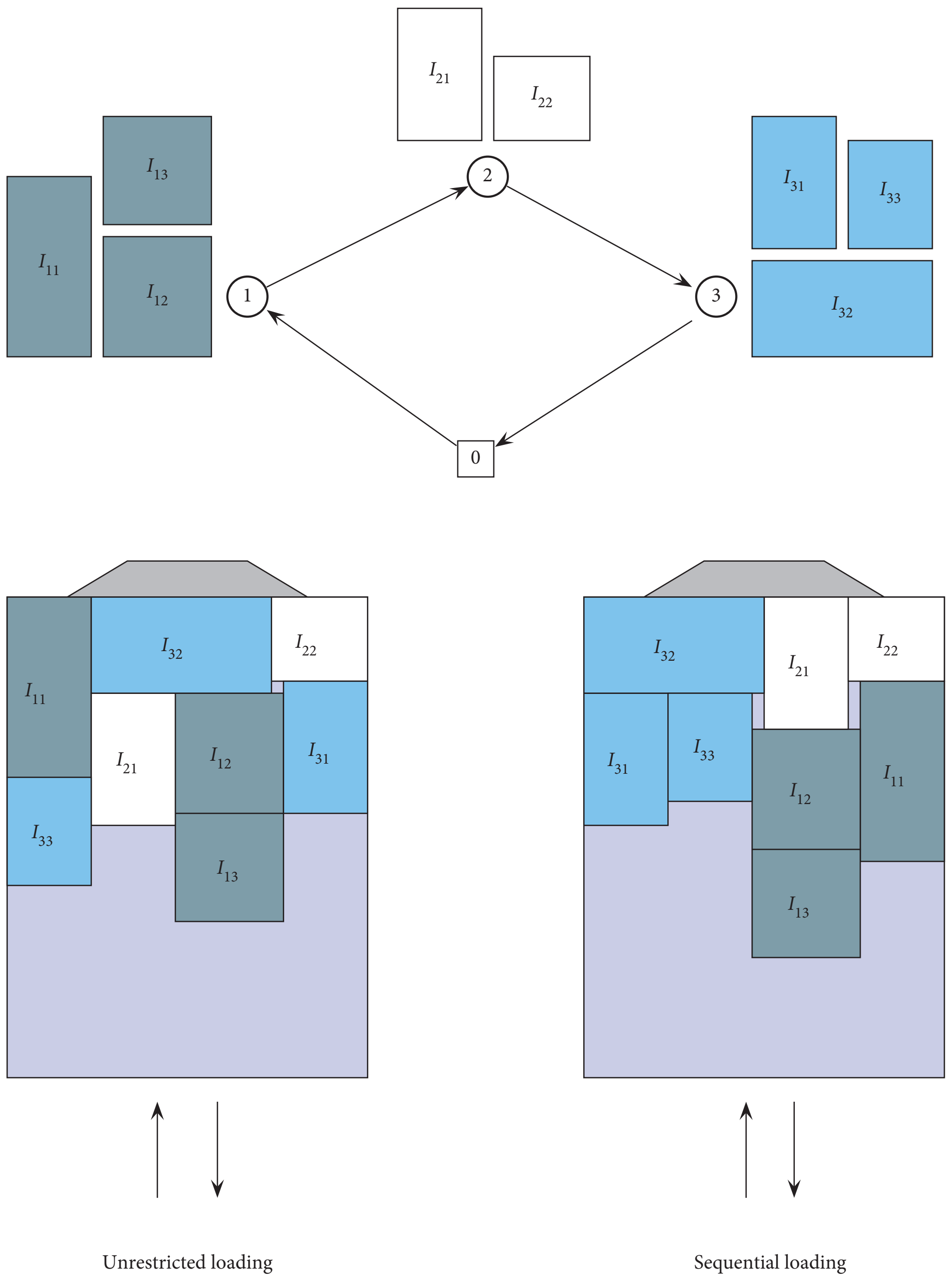

Figure 1: Unrestricted and sequential loading.

2L-HFVRP. Both these problems combine vehicle routing and two-dimensional loading constraints but consider a homogeneous and an unlimited heterogeneous fleet, respectively. As a generalization of VRP, these problems and their variants are known to be NP-hard problems, so exact methods are suitable for solving only small-scale problems, while metaheuristic algorithms are more favored for solving complex problems.

The 2L-CVRP was first presented by Iori et al. [8]. They addressed the sequential oriented version of the problem using an exact methodology that employs a branch-and-cut algorithm to deal with the routing aspects of the problem, combined with a nested branch-and-bound procedure to guarantee feasible loadings of the items into the vehicles. Subsequently, Gendreau et al. [9] proposed a metaheuristic based on tabu search for the routing problem, and checking for a feasible loading is performed via heuristics, lower bounds, and a truncated branch-and-bound procedure, considering both sequential and unrestricted oriented loading. Zachariadis et al. [10] developed a hybrid heuristic algorithm based on tabu search and guided local search and introduced a collection of packing heuristics to check the feasibility of the loading. Leung et al. [11] developed an improved algorithm based on guided tabu search and SA and introduced a new packing heuristic to address loading constraints. Fuellerer et al. [12] developed an ant colony algorithm to solve the routing problem, combined with heuristics for the loading subproblem. The authors addressed sequential and unrestricted problems, as well as oriented and nonoriented loading, and the rotation of items was allowed for the first time. Duhamel et al. [13] proposed a method that combines greedy adaptive random search with evolutionary local search and transformed the loading constraints into a project scheduling problem. A heuristic algorithm called promise routing-memory packing, based on the compression idea, was introduced by Zachariadis et al. [14]. Wei et al. [15] proposed an SA algorithm, with a mechanism of repeatedly decreasing and increasing the temperature that uses an open space-based heuristic to deal with loading constraints, which outperformed all previous approaches on all problem versions. Recently, Ji et al. [16] proposed an enhanced neighborhood search algorithm incorporated with a based tabu search packing algorithm to solve 2L-CVRP with split delivery. This variant was also addressed by Ferreira et al. [17] through an exact branch-and-cut approach, where a tailored procedure that includes the computation of lower bounds, a constructive-based heuristic, and a constraint programming model are proposed to handle the packing problem.

Concerning 2L-HFVRP, Leung et al. [18] first addressed the problem through an SA algorithm with a heuristic local search to solve the routing problem and a collection of six packing heuristics to solve the loading constraints, considering oriented sequential and unrestricted versions of the problem. Dominguez et al. [19] developed an algorithm that combines biased-randomized versions of routing and packing heuristics to solve oriented and nonoriented unrestricted problems. The biased randomization of heuristics refers to the use of skewed probability distributions to induce nonsymmetric random behavior in a heuristic procedure and transform a deterministic heuristic into a multistart probabilistic algorithm. A hybrid swarm algorithm based on artificial bee colony and artificial immune system was proposed by Zhang et al. [20] to deal with sequential and unrestricted problems. More recently, Sabar et al. [21] proposed an adaptive memetic algorithm that integrates new multiparent crossover operators with multilocal search algorithms in an adaptive manner for solving the routing problem, while hybridization of five packing heuristics is used to perform the loading process, considering only nonoriented unrestricted loading. A comparison between the previous studies, involving heterogeneous vehicle routing problems and two-dimensional loading constraints, and this study is shown in Table 1.

Our literature review of the vehicle routing problems that considers a heterogeneous fleet and two-dimensional loading constraints reveals that other researchers only solved these problems considering an unlimited fleet. However, in practice, it is not a realistic scenario, since companies generally own a limited fleet. Therefore, we address the 2L-HFFVRP.

Table 1: Contributions of previous studies.

Authors	Limited fleet	Items rotation	Sequential loading
Leung et al. [18]	No	No	Yes
Dominguez et al. [19]	No	Yes	No
Zhang et al. [20]	No	No	Yes
Sabar et al. [21]	No	No	No
This paper	Yes	No	Yes

3. Problem Statement

The 2L-HFFVRP is defined on a complete undirected graph $G = (N, A)$, where $N = \{0, 1, \ldots, n\}$ is a set of $n + 1$ vertices, the vertex 0 denotes the depot, and the vertices $1, 2, \ldots, n$ correspond to the positions of the customers $1, 2, \ldots, n$. $A = \{(i, j); i, j \in N\}$ is a set of edges, and each edge $(i, j) \in A$ is associated with a distance d_{ij} $(d_{ii} = 0)$ that denotes the distance from customer i to customer j. A set $V = \{1, \ldots, v\}$ of v vehicles is available at the depot. Each vehicle $k \in V$ has a weight capacity Q_k, a loading surface with length L_k and width W_k, a fixed cost F_k, and a variable cost V_k. In general, a vehicle with a larger capacity usually has higher fixed and variable costs. The traveling cost of each edge $(i, j) \in A$ by vehicle k is $C_{ij}^k = V_k \cdot d_{ij}$. Each customer i $(i = 1, \ldots, n)$ demands a set of m_i rectangular items, denoted by IT_i, and the total weight of IT_i is equal to D_i. Each item $I_{ir} \in IT_i$ $(r = 1, 2, \ldots, m_i)$ has a specific length l_{ir} and width w_{ir}.

For 2L-HFFVRP, a feasible solution must satisfy the following constraints:

(a) Each vehicle must start and finish at the depot

(b) Each customer can be served only once

(c) The weight capacity, length, and width of the vehicle cannot be exceeded

(d) All items of each customer must be loaded on the same vehicle

(e) Each item has a fixed loading orientation and must be loaded with its sides parallel to the sides of the loading surface

(f) Overlap between items within the same vehicle is not allowed

The objective of the 2L-HFFVRP is to find a set of routes of minimum cost that fulfill the constraints and satisfy all customers' demands. The constraints mentioned above refer to the unrestricted 2L-HFFVRP. Besides this problem version, this study also considers the sequential 2L-HFFVRP.

4. Mixed Integer Linear Programming Formulation

This section proposes a mathematical formulation to represent the 2L-HFFVRP, considering the characteristics described in Section 3. The notation used in the model, including indexes, sets, parameters, and variables, is presented in Section 4.1. The mathematical model is presented in the following stages: initially, a model for the HFFVRP is introduced in Section 4.2; then, we present constraints for vehicle loading in Section 4.3; and subsequently, sequential loading constraints are developed in Section 4.4.

4.1. Notations. The mixed-integer linear programming model depends on the sets, parameters, and variables described in Table 2.

4.2. Mathematical Formulation for the HFFVRP. A number of formulations could be used to model the HFFVRP [22]. In this study, we present a formulation based on the time-dependent formulation of the traveling salesman problem for integrated routing and packing problems [23]. Although this is a heavy four-index formulation, it gives us information about the position of customers in each route, which is necessary to ensure that the sequential loading constraints are fulfilled.

It is defined that a vehicle arrives at each vertex in stage t. Moreover, it is assumed that each route starts at vertex 0 in stage $t = 0$ and finishes at vertex 0 in stage $t = n + 1$ if only one vehicle is used. Therefore, the set of possible values for t is $T = \{1, \ldots, n + 1\}$.

The (routing) decision variables z_{ij}^{kt} indicate whether vehicle k goes directly from vertex i to vertex j in stage t ($z_{ij}^{kt} = 1$) or not ($z_{ij}^{kt} = 0$). It is assumed that $d_{ii} = 0$, $(i, i) \in A$. The mathematical model for the HFFVRP is expressed as follows:

$$\text{Minimize} \sum_{k\in V}\sum_{j\in N\backslash\{0\}} F_k z_{0j}^{k1} + \sum_{i\in N}\sum_{j\in N}\sum_{k\in V}\sum_{t\in T} V_k d_{ij} z_{ij}^{kt}, \quad (1)$$

subject to

$$\sum_{j\in N}\sum_{k\in V}\sum_{t\in T} z_{ij}^{kt} = 1, \quad i \in N\backslash\{0\}, \quad (2)$$

$$\sum_{j\in N}\sum_{k\in V}\sum_{t\in T\backslash\{1\}} t z_{ij}^{kt} - \sum_{g\in N}\sum_{k\in V}\sum_{t\in T} t z_{gi}^{kt} = 1, \quad i \in N\backslash\{0\}, \quad (3)$$

$$\sum_{j\in N\backslash\{0\}} z_{0j}^{k1} \le 1, \quad k \in V, \quad (4)$$

$$\sum_{j\in N} z_{ij}^{k,t+1} - \sum_{g\in N} z_{gi}^{kt} = 0, \quad i \in N\backslash\{0\}, k \in V, t \in T\backslash\{n+1\}, \quad (5)$$

$$\sum_{i\in N\backslash\{0\}}\sum_{j\in N}\sum_{t\in T\backslash\{1\}} D_i z_{ij}^{kt} \le Q_k, \quad k \in V, \quad (6)$$

$$z_{ij}^{kt} \in \{0, 1\}, \quad (i, j) \in A, k \in V, t \in T. \quad (7)$$

In this formulation, the objective function (1) aims to minimize the total cost for the vehicles to visit all customers. Constraints (2) ensure that each customer is visited exactly once. Constraints (3) ensure the connectivity of each tour; that is, they ensure that if customer i is visited in stage t, then this customer has to be the starting point for some other customer in stage $t + 1$. Constraints (4) ensure that each vehicle leaves the depot at most once in stage 1. Constraints (5) ensure that if vehicle k travels from customer l to customer i in stage t, then in stage $t + 1$, the same vehicle k must travel from customer i to another customer j. Constraints (6) ensure that the weight capacity of the vehicles is not exceeded. Constraints (7) are the (routing) decision variable domain constraints. To ensure the validity of this formulation, we assume that $z_{0j}^{kt} = 0$, $\forall j \in N\backslash\{0\}, k \in V, t \in T\backslash\{1\}$; that is, the vehicles can leave the depot only in the first stage.

4.3. Two-Dimensional Loading Constraints. In this section, we show how two-dimensional loading considerations can be embedded into formulation (1)–(7) of the last subsection. These constraints address the geometrical loading aspects; that is, they ensure that items are packed completely inside the vehicles and that they do not overlap each other in each vehicle.

Two-dimensional loading constraints for routing problems have been addressed by Junqueira [23], who developed models for integrated routing and loading problems; however, the formulation does not favor the optimum use of the vehicle loading surface when sequential loading is considered since it contains a parameter that limits the number of length units allowed to exceed the frontal border of the items of customers already loaded in order to arrange the items of a customer that will be visited earlier in the route. Due to this limitation, we build on a formulation for the container loading problem [24] adapted to the two-dimensional case and to the vehicle routing context.

Let us create a set I of all items to be loaded in the vehicles as follows. The first m_1 elements of I are the items of customer 1, the next m_2 elements are the items of customer 2, and so on. Therefore, each item $r \in IT_i$ becomes an item $p = \sum_{j=1}^{i-1} m_j + r$, so each customer i demands a set of items $p \in I$ ($p = \sum_{j=1}^{i-1} m_j + 1, \ldots, \sum_{j=1}^{i} m_j$), that is, $IT_i = \{\sum_{j=1}^{i-1} m_j + 1, \ldots, \sum_{j=1}^{i} m_j\}$. Each item $p \in I$ has a specific length l_p and width w_p. The total number of items in set I is $B = \sum_{i=1}^{n} m_i$. We assume that the dimensions of items and vehicles are integers, that items can be placed only orthogonally in a vehicle, and that their orientation is fixed.

A Cartesian coordinate system with its origin in the vehicle's front-left corner is adopted, and the vehicle door through which loading and unloading operations are performed is placed on the side between coordinates $(L, 0)$ and (L, W), as shown in Figure 2.

The (loading) decision variables indicate coordinates (x_p, y_p) of the front-left corner of item p in the vehicle loading surface, and the variables s_{pk} indicate whether item p is loaded into vehicle k ($s_{pk} = 1$) or not ($s_{pk} = 0$). Binary variables α_{pq}, β_{pq}, γ_{pq}, and δ_{pq} ($p, q \in I, p < q$) indicate the placement of items relative to each other. α_{pq}, β_{pq}, γ_{pq}, and δ_{pq} are equal to 1 if item p is, respectively, loaded behind, in

Table 2: Model sets, parameters, and variables.

Indexes/sets	
$i, j, g \in N = \{0, \ldots, n\}$	Set of vertices from the depot ($n = 0$) to customers ($n > 0$)
$k \in V = \{1, \ldots, v\}$	Set of vehicles
$t \in T = \{1, \ldots, n+1\}$	Set of stages in which a customer can be served by a vehicle
$p, q \in I = \{1, \ldots, \sum_{i=1}^{n} m_i\}$	Set of items demanded by all customers. Each customer i demands a set $IT_i = \left\{\sum_{j=1}^{i-1} m_j + 1, \ldots, \sum_{j=1}^{i} m_j\right\}$
Parameters	
n	Number of customers
d_{ij}	Distance from each customer i to each customer j
v	Number of vehicles
Q_k	Weight capacity of each vehicle k
L_k	Length of the loading surface of each vehicle k
W_k	Width of the loading surface of each vehicle k
F_k	Fixed cost of each vehicle k
V_k	Variable cost of each vehicle k
m_i	Number of items demanded by each customer i
D_i	Total weight demanded by each customer i
l_p	Length of each item p
w_p	Width of each item p
Variables	
z_{ij}^{kt}	Binary variable that indicates whether vehicle k goes directly from vertex i to vertex j in stage t ($z_{ij}^{kt} = 1$) or not ($z_{ij}^{kt} = 0)'$
x_p	Continuous variable that indicates the x-coordinate of the front-left corner of item p on the vehicle loading surface
y_p	Continuous variable that indicates the y-coordinate of the front-left corner of item p in the vehicle loading surface
s_{pk}	Binary variable that indicates whether item p is loaded into vehicle k $(s_{pk} = 1)'$ or not $(s_{pk} = 0)'$
α_{pq}	Binary variable that indicates whether item p is loaded behind item q $(\alpha_{pq} = 1)'$ or not $(\alpha_{pq} = 0)'$
β_{pq}	Binary variable that indicates whether item p is loaded in front of item q $(\beta_{pq} = 1)'$ or not $(\beta_{pq} = 0)'$
γ_{pq}	Binary variable that indicates whether item p is loaded on the left side of item q $(\gamma_{pq} = 1)'$ or not $(\gamma_{pq} = 0)'$
δ_{pq}	Binary variable that indicates whether item p is loaded on the right side of item q $(\delta_{pq} = 1)'$ or not $(\delta_{pq} = 0)'$

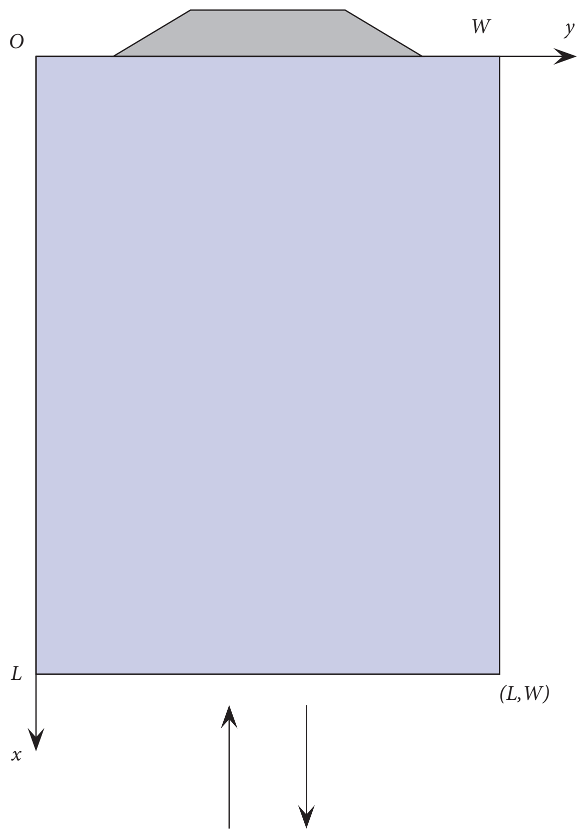

Figure 2: Vehicle loading surface.

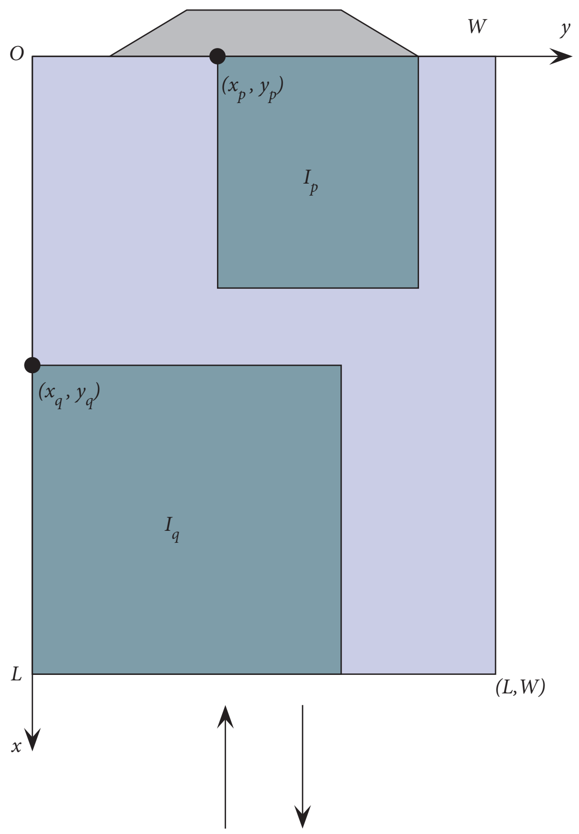

Figure 3: Relative positions of two items.

front of, on the left side of, or on the right side of item q. An example of the use of these variables is shown in Figure 3, which shows two items p and q loaded in a vehicle.

Assuming that $p < q$, we have $\alpha_{pq} = 1$, $\beta_{pq} = 0$, $\gamma_{pq} = 0$, and $\delta_{pq} = 0$. Note that an item p is said to be loaded behind or in front of an item q only if there is no intersection of its projections on the x-axis. Analogously, item p can be on the left or on the right of item q if there is no intersection of its projections on the y-axis. Therefore, it is possible to have $\alpha_{pq} + \beta_{pq} \le 1$ and $\gamma_{pq} + \delta_{pq} \le 1$, while $1 \le \alpha_{pq} + \beta_{pq} + \gamma_{pq} + \delta_{pq} \le 2$.

Let M be a sufficiently large number. The formulation for the 2L-HFFVRP is composed of the objective function (1) and constraints (2), (3), (4), (5), (6), and (7) presented before, plus the following constraints:

$$x_p + l_p \le x_q + (1 - \alpha_{pq})M, \quad p, q \in I, p < q, \tag{8}$$

$$x_q + l_q \le x_p + (1 - \beta_{pq})M, \quad p, q \in I, p < q, \tag{9}$$

$$y_p + w_p \le y_q + (1 - \gamma_{pq})M, \quad p, q \in I, p < q, \tag{10}$$

$$y_q + w_q \le y_p + (1 - \delta_{pq})M, \quad p, q \in I, p < q, \tag{11}$$

$$\alpha_{pq} + \beta_{pq} + \gamma_{pq} + \delta_{pq} \ge s_{pk} + s_{qk} - 1, \quad p, q \in I, p < q, k \in V, \tag{12}$$

$$x_p + l_p \le L_k + (1 - s_{pk})M, \quad p \in I, k \in V, \tag{13}$$

$$y_p + w_p \le W_k + (1 - s_{pk})M, \quad p \in I, k \in V, \tag{14}$$

$$\sum_{p \in IT_i} s_{pk} = m_i \sum_{j \in N} \sum_{t \in T} z_{ij}^{kt}, \quad i \in N \backslash \{0\}, k \in V, \tag{15}$$

$$x_p, y_p \in \mathbb{R}, \quad p \in I, \tag{16}$$

$$s_{pk} \in \{0, 1\}, \quad p \in I, k \in V, \tag{17}$$

$$\alpha_{pq}, \beta_{pq}, \gamma_{pq}, \delta_{pq} \in \{0, 1\}, \quad p, q \in I, p < q. \tag{18}$$

Constraints (8)–(11) ensure that the items do not overlap with each other. This check for overlap is necessary only if a pair of items is placed in the same vehicle, and this is ensured by constraints (12). Constraints (13) and (14) ensure that all the items loaded in a vehicle fit within the physical dimensions of the vehicle. Constraints (15) ensure the coupling of routing and loading structures; that is, all items required by customer i must be loaded on the same vehicle k that visits this customer. Finally, constraints (16)–(18) are the (loading) decision variable domain constraints.

4.4. Sequential Loading Constraints. In sequential loading problems, vehicle loading must take into account the delivery route of the vehicle and the sequence in which items are unloaded. In other words, items must be loaded in the reverse order in which respective customers are visited. This rule prevents unnecessary additional handling when each delivery point of the route is reached and, consequently, additional time spent unloading and reloading items of the remaining customers.

To address the sequential loading constraints and ensure that the items of a customer do not block the items of customers that will be served before them, we develop the following constraints:

$$x_q + l_q \le x_p + \left(2 + (\gamma_{pq} + \delta_{pq}) - \left(\sum_{g \in N} z_{gj}^{kt_2} + \sum_{g \in N} z_{gi}^{kt_1}\right)\right)M, \tag{19}$$
$$i, j \in N \backslash \{0\}, i < j, p \in IT_i, q \in IT_j, k \in V, t_1, t_2 \in T, t_1 < t_2,$$

$$x_p + l_p \le x_q + \left(2 + (\gamma_{pq} + \delta_{pq}) - \left(\sum_{g \in N} z_{gi}^{kt_2} + \sum_{g \in N} z_{gj}^{kt_1}\right)\right)M,$$
$$i, j \in N \backslash \{0\}, i < j, p \in IT_i, q \in IT_j, k \in V, t_1, t_2 \in T, t_1 < t_2. \tag{20}$$

If customer i is served before customer j, constraints (19) prevent the items of customer j from blocking the unloading of the items of customer i. These constraints verify whether customer i is served in stage t_1 preceding stage t_2 in which customer j is served. A particular case is illustrated in Figure 4 that shows an item p of customer i and an item q of customer j. If customer i is not visited before customer j, item p of customer i and item q of customer j have unrestricted x coordinates. However, if customer i is served before customer j, there are two possibilities: if item p of customer i is completely on the left or on the right side of item q of customer j, we have $\gamma_{pq} + \delta_{pq} = 1$, and there is no intersection of their projections on the y-axis, so the item does not block the other one. On the other hand, if item p is not completely on the left or right side of item q, an item is behind or in front of the other, and the constraints impose the condition that item p must be in front of item q. A similar explanation is valid for constraints (20) in the case that customer i is served after customer j.

The complete formulation for the 2L-HFFVRP with sequential loading constraints is given by the objective function (1) with the constraints (2)–(20). The four-index formulation is essential to properly model the 2L-HFFVRP when sequential loading constraints are present. Both indices k and t are fundamental to the modeling of the problem: index k is necessary to determine in which vehicle a given item is placed, and index t, in turn, makes it possible to indicate the position of a customer in the route and set the sequential loading constraints. The way these constraints are established allows us to obtain the optimal usage of vehicle loading surfaces, even for strongly heterogeneous items, once the position of a customer is analyzed in relation to the position of all other customers, not only those that immediately precede or follow the respective customer. If these constraints considered only the pairs of customers that are served in sequence in the route, there would be no guarantee that loading constraints would be satisfied.

This formulation can be used to optimally solve small-scale 2L-HFFVRP cases. A metaheuristic algorithm for handling large-scale problems is presented in the next section.

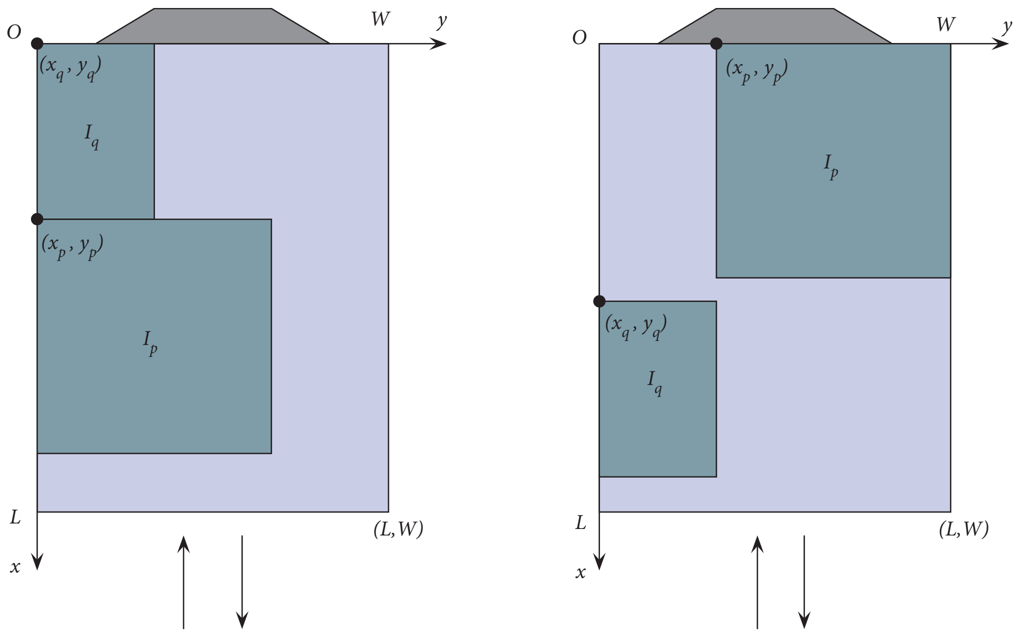

FIGURE 4: Relative positions of the items of two customers.

5. Hybrid SA Algorithm

Most approaches for solving problems that integrate vehicle routing and loading separate the problems into the following two stages: (1) a main algorithm for routing and (2) a tool for the loading per vehicle. In this paper, we present a hybrid algorithm based on SA and local search for the routing problem that uses heuristic methods to determine the geometric loading of items into vehicles. These heuristics are described, and the proposed algorithm is explained.

5.1. Heuristics for Two-Dimensional Loading. Given a route, it is necessary to determine whether it is feasible, that is, whether customer demand does not exceed vehicle capacity and whether all the items ordered by customers can be loaded onto the vehicle without overlap. In addition, in sequential loading, the item loading order must respect the order of customers served. To verify the feasibility of a route, we first examine whether the vehicle capacity is exceeded and whether all the items exceed the loading area and surface dimensions. If all conditions are satisfied, then heuristic methods are subsequently applied to determine whether feasible loading exists and, if it does, to determine its configuration.

Eight heuristics Heur$_i$ ($i = 1, \dots, 8$) are used to solve the two-dimensional loading problem; one item at a time is inserted in the most appropriate position in a list of available loading positions according to certain criteria to be described later in this section. Initially, only the left front corner of the vehicle loading surface, corresponding to the coordinate point (0,0), is available for loading an item. When an item is inserted, the position it occupies is excluded from the list of available loading positions, while at most two new positions are generated and added to this list. Figure 5 shows the mechanism of item insertion: item D is inserted in the highlighted position in the figure on the left, and the set of available positions after its insertion are shown in the figure on the right.

In sequential loading, when all items of a customer are loaded, positions that cannot be occupied by items that have not yet been loaded are excluded from the list of available loading positions Suppose that the route is $0 - i_1 - i_2 - i_3 - 0$. The partial loading for this route and the available loading positions are shown in Figure 6. After the loading of the items of customer i_2 and before the loading of the items of customer i_1, the highlighted positions are excluded from the list of available loading positions, as they would generate loadings that would contravene the sequential loading constraints.

Heuristics Heur$_i$ ($i = 1, \dots, 5$) are based on the study by Zachariadis et al. [10] and adopt the following criteria to determine an item loading position.

Heur$_1$: Bottom-Left Fill (x-axis): the selected position is that with the minimum x-axis coordinate, where ties are broken by the minimum y-axis coordinate. With this heuristic, packing tends to form strips parallel to the x-axis.

Heur$_2$: Bottom-Left Fill (y-axis): the selected position is that with the minimum y-axis coordinate, where ties are broken by the minimum x-axis coordinate. With this heuristic, packing tends to form strips parallel to the y-axis.

Heur$_3$: Max Touching Perimeter: the selected position is that with the maximum sum of the common edges between the item to be inserted, the loaded items in the vehicle, and the vehicle loading surface. This heuristic tends to spread items to the edges of the loading surface and later fills the inner part of it.

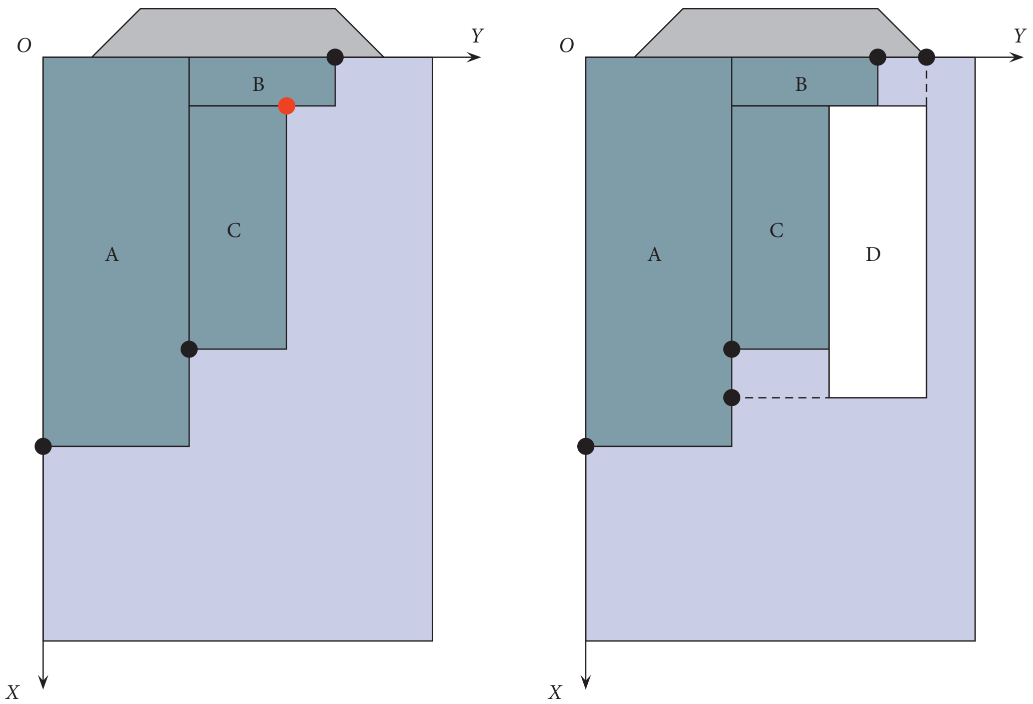

Figure 5: Process of loading items.

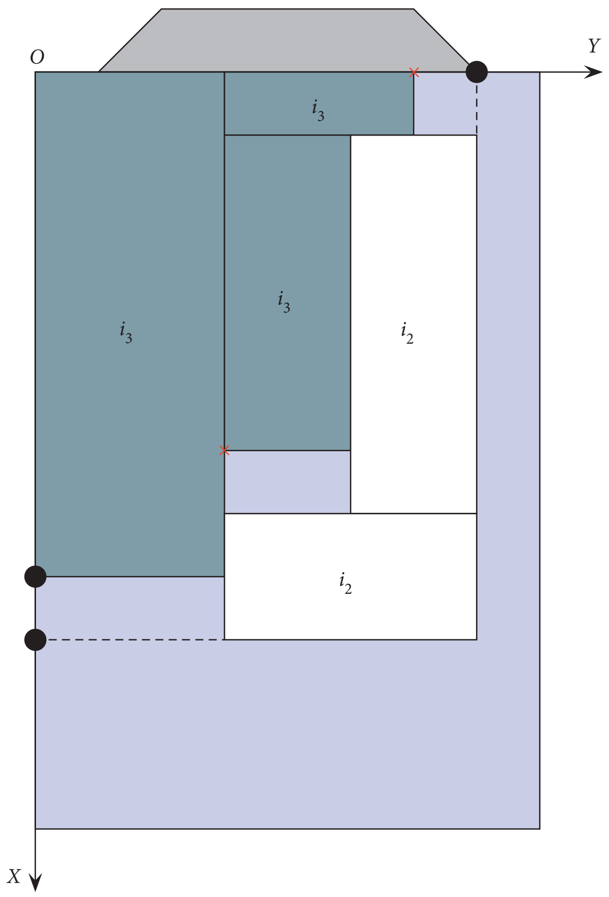

Figure 6: Infeasible positions for sequential loading.

Heur_4: Max Touching Perimeter no Walls: the selected position is that with the maximum sum of the common edges between the item to be inserted and the loaded items in the vehicle. This heuristic tends to fill the inner part of the loading surface before filling the edges.

Heur_5: Min Area: the selected position is that with the minimum rectangular surface, which is the rectangular area available for loading an item into this position.

A detailed explanation of these five heuristics can be found in [10]. As in the study by Zachariadis et al. [10], we first use these five packing heuristics to derive the loading strategy. Then, we introduce three new packing heuristics, namely, the Min Occupied Rectangular Area (Heur_6), Max Touching Perimeter Select Item (Heur_7), and Max Relative Touching Perimeter Select Item (Heur_8). Heur_6 selects an item loading position according the following criteria.

Heur_6: Min Occupied Rectangular Area: the selected position yields the minimum occupied rectangular area after item insertion. Figure 7 shows the occupied rectangular area after inserting item C in the given position. This strategy makes the packing compact.

Heuristics Heur_i $(i = 1, \ldots, 6)$ presented thus far employ individual criteria to select the position to insert a predefined item; that is, the items are loaded one at a time according to a given sequence. To increase the probability of the heuristics obtaining a feasible solution, three orders are generated for sequential and unrestricted cases. In a given route, each customer has a

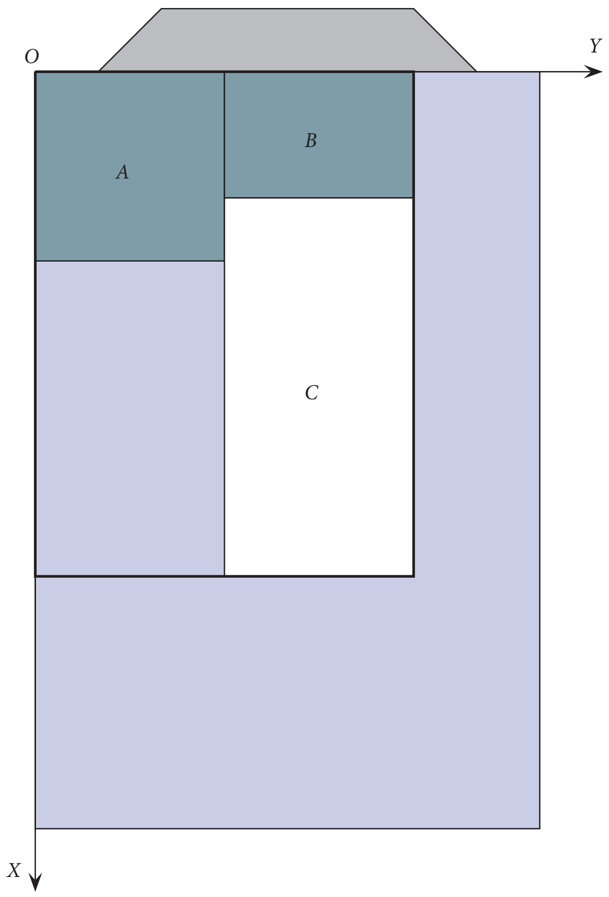

Figure 7: Occupied rectangular area in the heuristic Min Occupied Rectangular Area.

unique visit order. If sequential loading is considered, Ord_Seq$_1$, Ord_Seq$_2$, and Ord_Seq$_3$ are produced by sorting all items by the reverse customer visit order, breaking ties by decreasing length, width, and surface area, respectively. For the unrestricted problem, Ord_Un$_1$, Ord_Un$_2$, and Ord_Un$_3$ are produced simply by sorting all items by decreasing length, width, and surface area, respectively. Note that although Ord_Seq$_1$, Ord_Seq$_2$, and Ord_Seq$_3$ are primarily designed to deal with the sequential problem, they succeed in producing feasible unrestricted loadings in numerous cases when Ord_Un$_1$, Ord_Un$_2$, and Ord_Un$_3$ fail; therefore, they are also considered for solving unrestricted problems.

Heuristics Heur$_7$ and Heur$_8$ do not order items for loading, as they analyze the loading of all possible items to be loaded in all available loading positions that are feasible and select an item and a position that best meet the stated criteria. In sequential loading, the analysis is performed for all items of the current loading customer, while in unrestricted loading, any item can be inserted at any time. The criteria employed by Heur$_7$ and Heur$_8$ are as follows.

Heur$_7$: Max Touching Perimeter Select Item: for each item that can be loaded and for all the respective available loading positions that are feasible, the touching perimeter is calculated, as in Heur$_3$. The combination of an item and a position that matches the largest perimeter is chosen, and the item is inserted.

Heur$_8$: Max Relative Touching Perimeter Select Item: for each item that can be loaded and for all the respective available loading positions that are feasible, the relative touching perimeter is calculated as the ratio between the item perimeter and its touching perimeter. The combination of item and position that matches the largest ratio is selected, and the item is inserted.

The feasibility of a route is tested by heuristics Heur$_1$ to Heur$_8$, in this sequence, such that heuristics Heur$_i$ ($i = 1, \ldots, 6$) are executed for the orders mentioned earlier according to the sequential or unrestricted case. If any heuristic (considering any suitable order) finds a feasible loading for the route, the route is considered feasible, and the subsequent heuristics are not attempted. Otherwise, if none of the heuristics find a feasible loading for the route, the route is considered unfeasible. For each loading route feasibility check, the proposed packing heuristics are used. As these heuristics are computationally intensive, we use a data structure to avoid duplicate examinations.

5.2. Initial Solution. The initial solution is the starting point of the improvement metaheuristic proposed in the following subsections. The limited number of vehicles available in the depot and their capacities make it relatively difficult to determine an initial feasible solution. If at least one of the following situations occurs, a vehicle cannot service a customer: (1) the total weight of all customer items exceeds vehicle capacity; (2) the total area of all customer items exceeds the loading surface area; or (3) the length/width of any customer item exceeds the length/width of the loading surface. Owing to these limitations, the initial solution is constructed by prioritizing customers that can be served only by larger vehicles.

Given v vehicles available in the depot, it is assumed that $Q_1 \le Q_2 \le \cdots \le Q_v$, $F_1 \le F_2 \le \cdots \le F_v$, $V_1 \le V_2 \le \cdots \le V_v$, $L_1 \le L_2 \le \cdots \le L_v$, and $W_1 \le W_2 \le \cdots \le W_v$. First, for each vehicle k, a subset $CustomersList_k$ of customers that must be served by vehicle k or a larger vehicle is defined. Subsequently, for each vehicle $k = v, \ldots, 1$, the customers of set $CustomersList_k$ are added to a vehicle route $k, \ldots, v$ according to the minimum cost insertion so that the route is feasible. The cost of adding customer i to vehicle route k depends on whether route k is empty. If route k is empty, the cost is based on the fixed cost of vehicle k, plus the cost of traveling from the depot to customer i and back. If the route is not empty, the cost of adding customer i to all route positions is analyzed, and the minimum cost insertion is performed. If it is not possible to insert all the customers of some set $CustomersList_k$ in a particular route $k, \ldots, v$, the sequence of customer insertion is perturbed by randomly selecting a customer and a vehicle that can serve this customer to initialize the first route, and the procedure is restarted. Algorithm 1 provides a pseudocode for constructing the initial solution. $ToInsert_k$ is the set of customers of $CustomersList_k$ that have not yet been inserted in any route. At the beginning of route construction, $ToInsert_k = CustomersList_k$, and in the course of the algorithm, this set is updated as customers are added to routes.

5.3. Neighborhood Structures. SA explores the search space by performing moves to step from a current solution to a subsequent solution. The neighborhood structures are determined such that each movement causes an interroute disturbance in the solution. Intraroute improvement is accomplished by a local search procedure introduced in the next section.

Three types of movements are employed to disturb a solution, each of which defines a neighborhood structure, NS_1, NS_2, or NS_3, selected randomly in each loop with equal probability. Move type 1-Inter (NS_1) performs a customer relocation move [25]. It reassigns a customer from their current route to a position on another route (Figure 8). This move type can make some routes empty, reducing the number of vehicles used.

Move type 2-inter (NS_2) performs route exchange [25]. It swaps the positions of the two customers on different routes (Figure 9).

Move type 3-inter (NS_3) is a variant of route interchange 2-opt [26]. Two customers of different routes are selected; in each route, a block is created from the selected customer to the last customer, and these blocks are swapped between routes (Figure 10). For this move type, a fictitious customer can be selected in one of the two routes to shift a block of customers from one route to another. In this case, this move type can also empty some routes and reduce the number of vehicles used.

5.4. Local Search Mechanism. The neighborhood structures presented in Section 5.3 are employed by the SA algorithm to perform interroute improvements. In each new feasible solution found, at least one route has its set of customers modified; therefore, a local search method is used to optimize each modified route.

Given a single route, the local search performs exhaustive moves in its neighborhood until no improvement in solution can be achieved. Three move types, similar to those for interroute improvement, define neighborhood structures as intraroute optimization. Move type 1-Intra consists of the reallocation of a customer [25]; this move transfers a customer from one position to another on the same route (Figure 11). Move type 2-Intra swaps two customers on the route [25] (Figure 12). Finally, move type 3-Intra is defined by a 2-opt edge exchange method [26] that swaps the positions of two edges of the route (Figure 13).

Move types 1-intra, 2-intra, and 3-intra are, one at a time and in this order, applied to the route to perform all possible moves until no further improvement in route cost is found. This procedure is applied to the routes of the initial solution and every time interroute moves find a feasible solution during SA to examine other configurations of the new route.

5.5. SA Algorithm. SA is an algorithmic approach for solving combinatorial optimization problems [27, 28] that has been widely applied to vehicle routing problems. The proposed SA uses neighborhood structures NS_1, NS_2, and NS_3 to generate a candidate solution S' from a current solution S; this solution is accepted as the new current solution if it is better than S.

To avoid local optima, a worse solution may be accepted subject to the acceptance probability function $p(T, \Delta) = \exp(-\Delta/T)$, where $\Delta = \cos t(S') - \cos t(S)$ and T is a parameter of SA called temperature, which decreases during the process according to $T_k = 0.9 \cdot T_{k-1}$ to enforce the convergence of the search. At the beginning of the algorithm, T is assigned an initial value T_0. As suggested in [29, 30], $T_0 = -\Delta_{\max}/\ln(p_0)$ is defined to make any worst solution that leads to a cost increase $\Delta_{\max}$ accepted with a fixed probability p_0. To estimate $\Delta_{\max}$, move types 1-Inter, 2-Inter, and 3-Inter are randomly applied to the initial solution, and the value of $\Delta_{\max}$ is then estimated by the maximum absolute difference observed in the costs of two neighboring solutions.

For each temperature, a sequence of *Len* moves is carried out to explore the search space. If *Len* is too small, the solution space is not fully explored, whereas if *Len* is too large, the computational time required is rather long. Only feasible solutions are considered in our algorithm. Therefore, owing to the difficulty in finding feasible solutions to some instances of the problem, a maximum number of attempts $L = 10 \cdot Len$ is established for each temperature. For each feasible solution found, the local search procedure is employed to improve the routes that have been changed. To accelerate the speed of the algorithm, a condition is set such that if it finds a solution that is better than the best solution, that solution is replaced, and the current temperature is updated. The algorithm ends when the current temperature is lower than 0.01. Algorithm 2 provides a framework for the proposed SA methodology.

6. Computational Experiments

This section reports the results of computational experiments performed with the proposed mathematical formulation and the SA algorithm. Since we did not find any instances for the 2L-HFFVRP, we defined a set of instances by extending to the 2L-HFFVRP some 2L-HFVRP instances from the literature [18]. First, a detailed discussion of the benchmark instances is provided. Subsequently, the simulation results of benchmark instances are presented.

6.1. Benchmark Instances. To verify the effectiveness of the mathematical model and of the SA algorithm, small-scale and large-scale instances were used to estimate the results. These instances are described in this section.

Iori et al. [8] and Gendreau et al. [9] proposed a dataset including 36 2L-CVRP instances, in which the number of customers varies from 15 to 255 and each instance has five classes. In class 1, each customer demands one item with unit width and length, so the problems of this class are

```
for each customer $i$ do
  $k = 1$;
  while customer $i$ cannot be served by vehicle $k$ do
    $k = k + 1$;
end while
  insert customer $i$ into set $CustomersList_k$;
end for
for $k = 1$ to $v$ do
  $ToInsert_k \leftarrow CustomersList_k$;
end for
Route construction:
for $k = v$ to 1 do
  while $ToInsert_k$ is not empty do
    if it is possible to insert some customer of set $ToInsert_k$ into any route $R_j$ $(j = k, \ldots, v)$ then
      Execute the feasible insertion of customer $i$ into route $R_j$, which minimizes the insertion cost;
      Delete customer $i$ from the set $ToInsert_k$;
    else
      for $j = 1$ to $v$ do
        $ToInsert_j \leftarrow CustomersList_j$;
        Empty route $R_j$;
      end for
      Randomly select a vehicle $j = 1, \ldots, v$ and a customer $i \in ToInsert_j$;
      Insert customer $i$ into route $R_j$;
      Delete customer $i$ from the set $ToInsert_j$;
      go to Route construction;
    end if
  end while
end for
return generated solution
```

Algorithm 1: Pseudocode for constructing an initial solution.

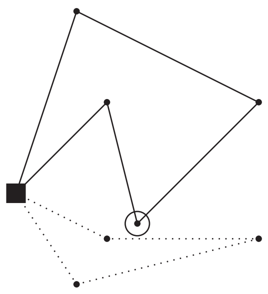
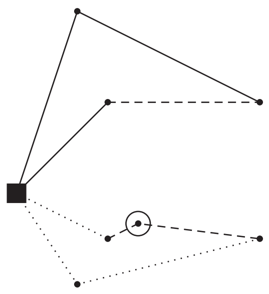
Figure 8: Interroute relocation move.

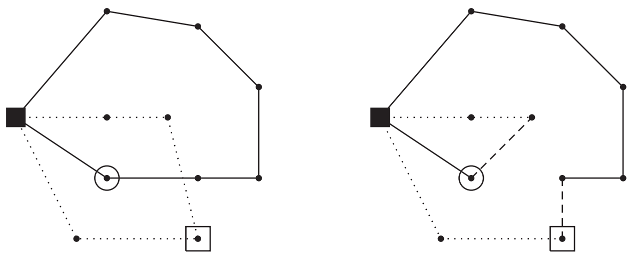
Figure 10: Interroute interchange move.

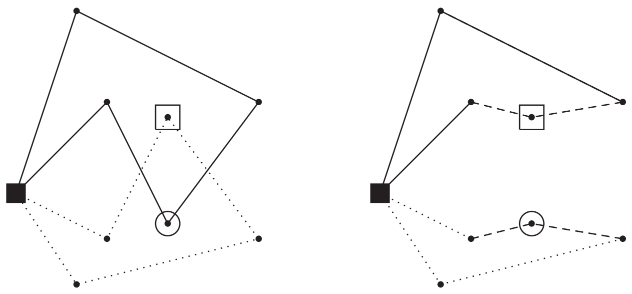
Figure 9: Interroute exchange move.

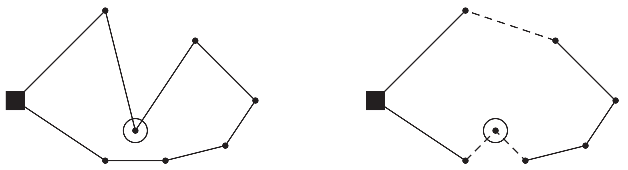
Figure 11: Intraroute relocation move.

similar to the pure CVRP. In classes 2–5, for each customer i, a set of m_i items with uniform distribution in the range (1, class number) is created. Each of these items is randomly classified, with an equal probability, into one of three possible shapes, namely, vertical, homogeneous, and horizontal, and the dimensions of the items are randomly generated in a given range. Leung et al. [18] modified 2L-CVRP instances to create 2L-HFVRP instances. The data for each benchmark were generated by eliminating the limit on the number of vehicles and adding information about

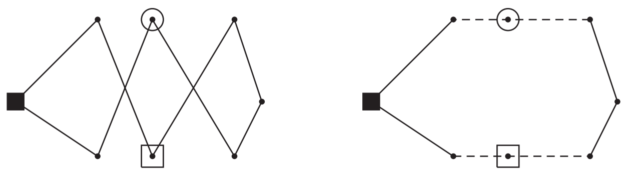
Figure 12: Intraroute exchange.

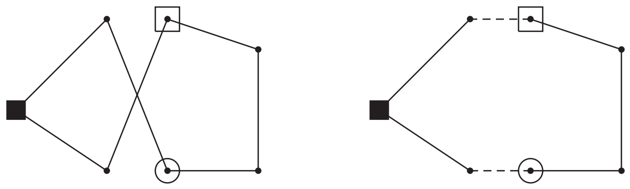
Figure 13: Intraroute interchange move.

```
Input: problem data; SA parameters: $p_0$; $Len$
Output: best solution $S^*$ found by the algorithm
Construct the initial solution $S_0$;
Apply the local search mechanism to $S_0$;
/* setting $T_0$ */
$S \leftarrow S_0$, $\Delta_{\max} = 0$, $i = 1$, $j = 1$;
while $i < Len$ and $j < L$ do
  Randomly select $NS$ from $\{NS_1, NS_2, NS_3\}$, and obtain a solution $S' \in NS(S)$;
  if $S'$ is feasible then
    Apply the local search mechanism to $S'$;
    if $|\text{cost}(S') - \text{cost}(S)| > \Delta_{\max}$ then
      $\Delta_{\max} = |\text{cost}(S') - \text{cost}(S)|$;
    end if
    $i = i + 1$;
    $S \leftarrow S'$;
    if $\text{cost}(S) < \text{cost}(S_0)$ then
      $S_0 \leftarrow S$;
    end if
  end if
  $j = j + 1$;
end while
$T_0 = -\Delta_{\max}/\ln(p_0)$;
/* SA algorithm */
$S \leftarrow S_0$, $S^* \leftarrow S_0$, $T = T_0$, $i = 1$, $j = 1$;
while stop criteria are not met do
  while $i < Len$ and $j < L$ do
    Randomly select $NS$ from $\{NS_1, NS_2, NS_3\}$, and obtain a solution $S' \in NS(S)$;
    if $S'$ is feasible then
      Apply the local search mechanism to $S'$;
      if $\text{cost}(S') < \text{cost}(S)$ then
        $S \leftarrow S'$;
        if $\text{cost}(S) < \text{cost}(S^*)$ then
          $S^* \leftarrow S$;
        end if
      else
        Set $S \leftarrow S'$ with probability $p = \exp(-(\cos t(S') - \cos t(S))/T)$
      end if
      $i = i + 1$;
    end if
    $j = j + 1$
  end while
  $T = 0.9 \cdot T$, $i = 1$, $j = 1$;
end while
return $S^*$
```

Algorithm 2: Pseudocode of the SA methodology for the 2L-HFFVRP.

capacity, loading surface, and fixed and variable costs for each of the four types of vehicles considered, namely, A, B, C, and D. For the five classes of the same instance, the vehicle types are the same.

For 2L-HFFVRP, we generate benchmark data by limiting the number of vehicles of types A, B, C, and D. For the five classes of the same instance, the number of vehicles of each type was the same. Table 3 provides details on the number of vehicles of each type considered for each instance. Instances of class 1 correspond to the pure HFFVRP since every route is feasible in terms of the loading constraints.

To validate the mathematical formulation, 8 small-scale instances, namely, P3, P4, P5, P6, P7, P8, P9, and P10, are generated based on instance 1 for the 2L-HFFVRP. For classes 1–5, problems P3, P4, P5, P6, P7, P8, P9, and P10 take into account the first 3, 4, 5, 6, 7, 8, and 9 customers of instance 1 for the 2L-HFFVRP, respectively. Instances of class 1 correspond to the pure HFFVRP since every route is feasible in terms of the loading constraints.

6.2. SA Parameter Setting. The proposed SA algorithm has two parameters that need to be set by the user: the probability p_0 of accepting any worst solution at the initial temperature and the number *Len* of feasible moves to be performed at each temperature. To yield experimental results with better convergence, the parameters of SA are determined by conducting experiments on a set of large-scale instances using various parameter values, involving both unrestricted and sequential problem versions. The suggested parameter values, along with the tested ranges, are reported in Table 4.

6.3. Results and Discussion. The 2L-HFFVRP mathematical formulation of Section 3 and the SA algorithm of Section 4 were implemented in the Visual Basic .NET programming language. The mathematical model was solved by CPLEX 12.6 with the default configuration parameters. All the computational tests were executed in a machine with the following specifications: a 1.4 GHz Intel Core i5 processor with 8 GB RAM memory running a Windows 10 operating system. The pure HFFVRP was solved for problems of class 1. For problems of classes 2–5, the 2L-HFFVRP was solved regarding both unrestricted and sequential loading.

Small-scale problems were solved using the mathematical model and the SA algorithm, and the results are summarized in Tables 5–7. To evaluate the model performance, the time taken by CPLEX to solve each model was limited to 3600 seconds. With respect to the quality of the solution obtained by CPLEX, there were four possible cases: (i) the optimal solution is obtained; (ii) a nonoptimal solution is obtained, with CPLEX exceeding the time limit; (iii) no solution is obtained, with CPLEX exceeding the time limit; and (iv) there is insufficient computer memory to solve the model. The last two cases are represented in the tables by the symbol "–." Concerning SA, for each instance, five replications of the algorithm were executed, and the best results were obtained.

TABLE 3: Dataset for the 2L-HFFVRP instances.

Inst.	Vehicle type			
	A	*B*	*C*	*D*
1	0	2	3	2
2	0	0	3	3
3	0	3	5	
4	1	3	4	
5	0	2	2	2
6	0	4	5	
7	0	3	3	2
8	0	3	1	3
9	0	1	8	
10	0	5	4	2
11	0	1	5	3
12	0	6	14	
13	0	1	6	
14	1	3	3	4
15	0	5	5	4
16	0	10	7	
17	1	3	13	5
18	0	5	7	
19	1	4	8	8
20	1	7	14	6
21	2	6	12	11
22	1	6	8	12
23	0	8	10	11
24	0	4	9	13
25	0	10	12	15
26	1	14	8	15
27	1	6	18	14
28	2	9	17	16
29	0	17	24	
30	3	9	17	22
31	1	17	36	26
32	3	28	20	26
33	1	19	24	29
34	1	33	29	33
35	6	22	46	
36	7	33	46	9

The results for HFFVRP small-scale problems of class 1 are presented in Tables 5, which shows, for each instance, the number of customers, the solution cost and the runtime (in seconds) of CPLEX and SA for solving the problem, and the percentual difference between the solution costs of CPLEX and SA. Tables 6 and 7 show these results for the unrestricted and sequential 2L-HFFVRP, respectively, for small-scale problems of classes 2–5 and include the number of items to be loaded for each instance.

According to the results, the problems become more complex as the numbers of customers and items to be loaded increase. By comparing the results for the HFFVRP (Table 5) and for the unrestricted 2L-HFFVRP (Table 6), we notice that, for the same routing problem configuration, some solution costs deteriorate due to loading constraints. In fact, many routes become infeasible as these constraints are considered. By comparing the results for the unrestricted and sequential 2L-HFFVRP (Tables 6 and 7), we find that a few solution costs deteriorate when sequential loading constraints are embedded into the problem, mainly for problems with more customers and items.

Table 4: The parameter settings of SA.

Parameter	Tested range	Suggested value
Probability p_0 of accepting any solution at the initial temperature	0.05–0.20	0.10
Number *Len* of feasible moves to be performed at each temperature	5000–20 000	10 000

Table 5: Comparison of results of CPLEX and SA on small-scale HFFVRP of class 1.

Instance	Number of customers	CPLEX		SA		% gap
		Cost	Time (s)	Cost	Time (s)	
P3	3	130.25	0.27	130.25	29.65	0.00
P4	4	188.10	0.27	188.10	37.23	0.00
P5	5	222.35	0.50	222.35	33.55	0.00
P6	6	255.96	0.89	255.96	24.46	0.00
P7	7	308.36	2.75	317.51	26.97	2.97
P8	8	345.48	4.36	345.48	29.19	0.00
P9	9	394.97	16.70	394.97	31.03	0.00
P10	10	416.61	587.25	417.40	13.90	0.19
Average			76.62		28.25	0.39

Table 6: Comparison of results of CPLEX and SA on small-scale unrestricted 2L-HFFVRP of classes 2–5.

Instance	Class	Number of customers	Number of items	CPLEX		SA		% gap
				Cost	Time (s)	Cost	Time (s)	
P3	2	3	6	130.25	0.43	130.25	17.42	0.00
	3	3	6	130.25	0.32	130.25	32.17	0.00
	4	3	7	130.25	0.34	130.25	30.95	0.00
	5	3	12	130.25	0.55	130.25	38.95	0.00
P4	2	4	8	201.55	0.67	201.55	20.56	0.00
	3	4	7	188.10	0.54	188.10	27.06	0.00
	4	4	11	201.55	0.52	201.55	34.36	0.00
	5	4	13	201.55	0.46	201.55	39.67	0.00
P5	2	5	10	222.35	0.49	222.35	21.37	0.00
	3	5	9	237.52	0.54	237.52	12.98	0.00
	4	5	12	222.35	0.66	222.35	28.92	0.00
	5	5	17	222.35	1.71	222.35	32.52	0.00
P6	2	6	11	257.31	0.93	257.31	12.53	0.00
	3	6	11	257.31	1.17	257.31	24.44	0.00
	4	6	16	257.31	2.17	257.31	23.63	0.00
	5	6	18	257.31	3.37	257.31	16.13	0.00
P7	2	7	12	308.36	2.84	317.51	27.03	2.97
	3	7	12	308.36	3.69	318.81	12.44	3.39
	4	7	19	308.36	8.65	318.81	19.92	3.39
	5	7	23	308.36	13.38	317.51	22.64	2.97
P8	2	8	13	345.48	2.86	345.48	11.63	0.00
	3	8	15	371.76	8.27	371.76	13.86	0.00
	4	8	22	345.48	15.68	345.48	31.64	0.00
	5	8	26	345.48	46.06	345.48	17.40	0.00
P9	2	9	15	406.73	6.54	406.73	9.28	0.00
	3	9	17	397.47	64.76	397.47	26.43	0.00
	4	9	24	394.97	75.75	394.97	11.27	0.00
	5	9	30	394.97	996.45	397.46	6.98	0.63
P10	2	10	17	446.91	223.98	446.91	11.13	0.00
	3	10	29	442.88	3600.04	453.13	11.74	2.31
	4	10	25	422.04	2444.98	430.86	15.11	2.09
	5	10	33	430.86	3600.09	439.00	12.25	1.89
Average					347.78		21.08	0.61

Table 7: Occupied rectangular area in the heuristic Min Occupied Rectangular area.

Instance	Class	Number of customers	Number of items	CPLEX		SA		% gap
				Cost	Time (s)	Cost	Time (s)	
P3	2	3	6	130.25	0.31	130.25	9.46	0.00
	3	3	6	130.25	0.38	130.25	15.96	0.00
	4	3	7	130.25	0.44	130.25	21.46	0.00
	5	3	12	130.25	0.84	130.25	34.86	0.00
P4	2	4	8	201.55	0.54	201.55	9.28	0.00
	3	4	7	188.10	0.58	188.10	15.07	0.00
	4	4	11	201.55	1.81	201.55	20.67	0.00
	5	4	13	201.55	2.21	201.55	39.76	0.00
P5	2	5	10	222.35	2.34	222.35	9.56	0.00
	3	5	9	237.52	1.97	237.52	22.05	0.00
	4	5	12	222.35	5.53	222.35	22.77	0.00
	5	5	17	222.35	15.08	222.35	44.13	0.00
P6	2	6	11	257.31	9.99	257.31	10.08	0.00
	3	6	11	257.31	8.60	257.31	18.83	0.00
	4	6	16	257.31	28.48	257.31	21.25	0.00
	5	6	18	257.31	34.61	257.31	25.92	0.00
P7	2	7	12	308.36	20.75	317.51	14.33	2.97
	3	7	12	308.36	39.11	318.81	12.73	3.39
	4	7	19	308.36	137.70	326.82	17.81	5.99
	5	7	23	308.36	440.71	317.51	12.84	2.97
P8	2	8	13	345.48	44.89	345.48	11.40	0.00
	3	8	15	371.76	146.19	371.76	12.43	0.00
	4	8	22	345.48	653.90	345.48	15.17	0.00
	5	8	26	345.48	2207.99	345.48	20.63	0.00
P9	2	9	15	406.73	473.62	406.73	8.42	0.00
	3	9	17	397.47	986.25	397.47	9.44	0.00
	4	9	24	394.97	3600.24	*394.9	11.43	0.00
	5	9	30	429.88	3600.26	406.73	11.30	−5.39
P10	2	10	17	–	–	*446.9	10.46	–
	3	10	29	453.43	3600.20	453.13	9.81	−0.07
	4	10	25	477.25	3600.33	430.86	14.39	−9.72
	5	10	33	–	–	430.86	17.55	–
Average					655.53		17.23	0.004

Tables 5–7 reveal that when the numbers of customers and items to be loaded are small, CPLEX can quickly find an optimal solution. However, as the numbers of customers and items increase, the solution time of CPLEX increasingly lengthens. Note that CPLEX does not find the optimal solutions in the stated time for the unrestricted problem P10 of classes 3 and 5, for the sequential problem P9 of classes 4-5, or for any class of sequential problem P10.

The results provided by SA are very close to optimality. According to Table 5, the SA algorithm finds the optimal solution for 8 of the 10 HFFVRP instances of class 1, and the average gap is 0.39%. Table 6 shows that SA finds the optimal solution for 24 of the 32 unrestricted 2L-HFFVRP instances of classes 2–5 and that the average gap is 0.61%. Table 7 shows that for 24 out of the 32 sequential 2L-HFFVRP instances of classes 2–5, the optimal solution is obtained by SA. For instances P9 of class 4 and P10 of class 2 (marked with an “ $*$ ” in Table 7), the CPLEX results are not proven to be optimal by the software. Nevertheless, solutions obtained by SA for these instances of the sequential 2L-HFFVRP are equal to the proven optimal solutions obtained by CPLEX for the unrestricted 2L-HFFVRP. Therefore, we can say that the SA solutions for sequential instances P9 of class 4 and P10 of class 2 are optimal. In addition, for sequential problem P10 of class 5, CPLEX does not manage finding a solution in the stated time, but the solution obtained by SA is equal to the CPLEX solution for the unrestricted correspondent problem; thus, for this instance, we consider the SA solution to be very good, if not optimal. For three instances (P9 of class 5 and P10 of classes 2 and 3), the solutions obtained by SA are better than solutions found by CPLEX. The average gap between the SA algorithm and CPLEX is 0.004%. The SA computational time is shorter than the CPLEX computational time only for instances with more customers and items—namely, instance P10 of class 1; instances P9 and P10 for unrestricted problems; and instances P7, P8, P9, and P10 for sequential problems. Nevertheless, the average computational times of SA are 28.25 seconds for the HFFVRP, 21.08 seconds for the unrestricted 2L-HFFVRP, and 17.23 seconds for the sequential 2L-HFFVRP, against the 76.62 seconds, 347.78 seconds, and 655.53 seconds of CPLEX, respectively. Therefore, the SA algorithm is effective in solving problems with larger numbers of customers and items.

Table 8: Results of SA on large-scale instances.

	Class 1		Classes 2-5			
	HFFVRP		Unrestricted 2L-HFFVRP		Sequential 2L-HFFVRP	
Instance	Cost	Time (s)	Cost	Time (s)	Cost	Time (s)
1	683.78	10.43	715.77	10.48	719.50	22.27
2	716.75	13.47	758.24	11.28	760.17	21.61
3	800.24	15.83	877.29	12.02	881.60	26.54
4	728.59	13.10	791.68	10.01	797.62	24.14
5	782.52	14.39	931.02	11.76	1009.91	24.32
6	890.97	20.78	968.17	13.65	968.23	29.87
7	4012.22	27.66	6109.17	21.02	6233.76	57.40
8	4105.17	26.72	7267.27	23.01	7269.30	51.19
9	1130.41	14.43	1198.13	11.95	1206.95	24.24
10	5824.73	56.61	8454.78	24.58	8456.40	54.55
11	6168.48	56.19	9530.84	29.81	9591.70	61.07
12	1806.82	13.98	1822.95	10.96	1824.97	22.29
13	21 551.38	75.02	30 529.26	46.29	31 009.94	116.03
14	10 804.36	19.81	11 529.68	13.09	11 544.31	29.05
15	10 529.44	26.94	12 541.39	17.87	12 816.66	39.12
16	1454.31	12.41	1486.74	10.35	1506.31	22.80
17	2166.90	11.99	2232.44	9.92	2234.98	21.06
18	3464.58	60.79	6350.55	29.78	6652.65	82.37
19	2275.39	32.50	4735.34	24.37	4905.96	60.60
20	2387.60	47.24	6578.13	22.65	6793.63	50.48
21	4462.11	37.23	10 038.31	26.09	10 549.43	75.65
22	5678.09	29.66	10 997.60	25.58	11 487.42	68.85
23	5166.67	35.08	10 549.34	24.82	10 968.85	50.65
24	3803.35	23.25	5422.38	19.92	5769.50	54.52
25	5509.09	43.11	14 162.47	28.21	14 683.91	91.52
26	7658.30	55.08	15 482.92	20.98	16 300.15	61.13
27	3745.45	26.71	6799.65	21.61	6999.79	53.24
28	5053.22	74.74	25 484.50	29.33	27 079.76	121.11
29	11 004.28	113.23	27 967.09	46.65	28 522.72	202.55
30	8688.92	44.01	19 487.93	29.11	20 362.33	116.05
31	9799.88	33.43	23 957.03	32.68	24 363.94	293.83
32	11 043.77	22.61	24 775.07	27.17	25 988.46	157.52
33	11 065.01	44.18	25 983.19	35.33	26 493.88	118.10
34	7258.49	19.33	16 023.98	31.72	16 455.94	143.01
35	5367.14	68.90	11 148.20	54.88	11 596.82	598.67
36	3557.03	26.26	6226.45	43.11	6366.40	316.60
Average	5309.60	35.20	10 275.41	23.95	10 588.16	93.45

The large-scale problems described in Section 5.2 are solved by the SA algorithm. Table 8 presents the results for the HFFVRP instances of class 1 and the average results for the unrestricted and sequential 2L-HFFVRP instances of classes 2–5. The results of Table 8 reinforce the idea that loading constraints worsen the solution costs of the HFFVRP and that sequential loading produces slightly higher solution costs than the related unrestricted problem. The computational times are longer for sequential problems. On average, the time spent by SA to solve the sequential 2L-HFFVRPs is 4 times longer than the time spent solving the unrestricted 2L-HFFVRPs. In fact, computational experiments reveal that the major difficulty in solving the 2L-HFFVRP concerns finding feasible loadings. As the number of vehicles is limited, a reasonable part of the computational time is spent finding an initial feasible solution for some instances.

In this sense, the impact of the three new packing heuristics, $Heur_6$, $Heur_7$, and $Heur_8$, proposed in this study is analyzed. During the tests, the packing heuristic that produces each feasible loading is recorded. The ratios between feasible loadings found by heuristics $Heur_6$, $Heur_7$, and $Heur_8$ and all found feasible loadings for unrestricted and sequential problems are presented in Table 9. For instance, 3.11% of all the feasible loadings found during the tests is obtained by heuristic $Heur_7$. This means that no other previous heuristic can find these feasible loadings. Therefore, all three new heuristics are able to find new feasible loadings that cannot be obtained by the other previous heuristics. Heuristic $Heur_7$ stands out regarding managing more complex loadings in both unrestricted and sequential problems. For sequential problems, heuristic $Heur_6$ manages to obtain new feasible loadings. In fact, heuristic $Heur_6$ tends to form rectangular blocks of items, which favors sequential

Table 9: Percentage of the feasible loadings obtained by heuristics $Heur_6$, $Heur_7$, and $Heur_8$

Heuristic	$Heur_6$	$Heur_7$	$Heur_8$
Unrestricted 2L-HFFVRP	0.05	3.11	0.33
Sequential 2L-HFFVRP	1.28	1.96	0.45

loading of blocks of items of the same customer. Although heuristic $Heur_8$ is the last to be executed, the results in Table 9 indicate that this heuristic is able to find feasible solutions not found by any of the previous heuristics.

7. Managerial Insights

Based on our results, several managerial implications can be used to help logistic providers and managers in optimizing the routes and loading configurations of a heterogeneous fleet of vehicles. The results presented in the previous section have shown the efficiency and robustness of the SA algorithm in solving unrestricted and sequential 2L-HFFVRP. The results of this method can be effectively used in decision-making at the strategic and tactical levels, leading to better use of the fleet of vehicles and cost reduction.

There is a trade-off between solution cost and computational time. Better solutions can be obtained by increasing the values of parameters p_0 and *Len*, but the computational time of the algorithm will also increase. According to the period of time the company usually has to plan routes, it is possible to set the algorithm parameters to obtain better solutions within this time.

We compared sequential loading to unrestricted loading and observed that, on average, for classes 2–5 in 36 instances, the total transportation cost increased by around 3% when sequential loading is considered. Therefore, if rearranging other customers' items at each customer site is not a problem for the company, especially regarding the time these operations will take, unrestricted loading should be considered, as it implies lower cost solutions compared to sequential loading. Nevertheless, it is worth noticing that, in some cases, unloading and reloading operations are very time-consuming. For cases with up to 40 customers, we observed that the percentual difference between solution costs in unrestricted and sequential problems is, on average, smaller than that for cases with more customers. In these cases, the time saving of considering sequential loading is worth analyzing.

8. Conclusions

In this paper, the 2L-HFFVRP, a new variant of the VRP, is presented. In this problem, each customer demands a set of rectangular two-dimensional items, and the objective is to find the minimum cost delivery routes for a limited set of vehicles with different capacities, fixed and variable operating costs, and a rectangular two-dimensional loading surface. Both the unrestricted and the sequential versions of the problem are handled. This problem is interesting in terms of both theoretical complexity and real-world applications. To the best of our knowledge, the 2L-HFFVRP has not been previously addressed in the literature, and nor have the sequential loading constraints.

To solve this problem, a mixed integer linear programming model was formulated. Computational tests were performed with 10 small-scale instances regarding five classes of problems. The results show that the model is consistent and properly represents the problem treated and that this approach is able to solve problems where the numbers of customers and items are relatively small. Therefore, with the developed model and for small cases, it is possible to determine the optimal solutions, or in some cases a good lower bound, in order to be able to compare the results achieved with nonexact methods or have a base of comparison of their performances. Moreover, the proposed model can be useful for motivating future research exploring exact methods for solving 2L-HFFVRP.

Owing to the complexity of the problem model, a hybrid algorithm that involves SA and packing heuristics was proposed, and three new packing heuristics were developed. By testing the proposed framework on 36 large-scale instances, each within five classes, we verified that the proposed methodology can solve this difficult problem in an acceptable computational time; hence, the proposed SA algorithm can further be assessed to solve different variants of 2L-HFFVRP. The three new packing heuristics were also proven to be effective in finding feasible loadings not found by the other heuristics, thus increasing the probability of obtaining a feasible loading.

Our study has some limitations and can be extended in several aspects in future research. It would be reasonable to examine the more realistic scenarios that sometimes arise in practical applications. First, concerning the routing problem, future work can integrate practical constraints such as split delivery and time windows. Second, in terms of two-dimensional loading configuration, we examined only oriented problems. Thus, future research can consider items rotation as it could substantially improve vehicle occupation. Third, although the proposed SA algorithm for solving 2L-HFFVRP has obtained good solutions, there is still room for improvement, which may be achieved by proposing more sophisticated and superior routing and packing strategies.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

References

[1] S. Bhuniya, S. Pareek, and B. Sarkar, "A supply chain model with service level constraints and strategies under uncertainty," *Alexandria Engineering Journal*, vol. 60, no. 6, pp. 6035–6052, 2021.

[2] B. K. Dey, S. Bhuniya, and B. Sarkar, "Involvement of controllable lead time and variable demand for a smart manufacturing system under a supply chain management," *Expert Systems with Applications*, vol. 184, Article ID 115464, 2021.

[3] S.-Yi Tan and W.-C. Yeh, "The vehicle routing problem: state-of-the-art classification and review," *Applied Sciences*, vol. 11, no. 21, Article ID 10295, 2021.

[4] Ç. Koç, T. Bektaş, O. Jabali, and G. Laporte, "Thirty years of heterogeneous vehicle routing," *European Journal of Operational Research*, vol. 249, no. 1, pp. 1–21, 2016.

[5] M. Iori and S. Martello, "Routing problems with loading constraints," *Top*, vol. 18, no. 1, pp. 4–27, 2010.

[6] H. Pollaris, K. Braekers, A. Caris, G. K. Janssens, and S. Limbourg, "Vehicle routing problems with loading constraints: state-of-the-art and future directions," *Spectrum*, vol. 37, no. 2, pp. 297–330, 2015.

[7] J. F. Côté, G. Guastaroba, and M. G. Speranza, "The value of integrating loading and routing," *European Journal of Operational Research*, vol. 257, no. 1, pp. 89–105, 2017.

[8] M. Iori, J.-J. Salazar-González, and D. Vigo, "An exact approach for the vehicle routing problem with two-dimensional loading constraints," *Transportation Science*, vol. 41, no. 2, pp. 253–264, 2007.

[9] M. Gendreau, M. Iori, G. Laporte, and S. Martello, "A tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints," *Networks*, vol. 51, no. 1, pp. 4–18, 2008.

[10] E. E. Zachariadis, C. D. Tarantilis, and C. T. Kiranoudis, "A guided tabu search for the vehicle routing problem with two-dimensional loading constraints," *European Journal of Operational Research*, vol. 195, no. 3, pp. 729–743, 2009.

[11] S. C. H. Leung, X. Zhou, D. Zhang, and J. Zheng, "Extended guided tabu search and a new packing algorithm for the two-dimensional loading vehicle routing problem," *Computers & Operations Research*, vol. 38, no. 1, pp. 205–215, 2011.

[12] G. Fuellerer, K. F. Doerner, R. F. Hartl, and M. Iori, "Ant colony optimization for the two-dimensional loading vehicle routing problem," *Computers & Operations Research*, vol. 36, no. 3, pp. 655–673, 2009.

[13] C. Duhamel, P. Lacomme, A. Quilliot, and H. Toussaint, "A multi-start evolutionary local search for the two-dimensional loading capacitated vehicle routing problem," *Computers & Operations Research*, vol. 38, no. 3, pp. 617–640, 2011.

[14] E. E. Zachariadis, C. D. Tarantilis, and C. T. Kiranoudis, "Integrated distribution and loading planning via a compact metaheuristic algorithm," *European Journal of Operational Research*, vol. 228, no. 1, pp. 56–71, 2013.

[15] L. Wei, Z. Zhang, D. Zhang, and S. C. H. Leung, "A simulated annealing algorithm for the capacitated vehicle routing problem with two-dimensional loading constraints," *European Journal of Operational Research*, vol. 265, no. 3, pp. 843–859, 2018.

[16] B. Ji, S. Zhou, S. S. Yu, and G. Wu, "An enhanced neighborhood search algorithm for solving the split delivery vehicle routing problem with two-dimensional loading constraints," *Computers & Industrial Engineering*, vol. 162, Article ID 107720, 2021.

[17] K. M. Ferreira, T. A. de Queiroz, and F. M. B. Toledo, "An exact approach for the green vehicle routing problem with two-dimensional loading constraints and split delivery," *Computers & Operations Research*, vol. 136, Article ID 105452, 2021.

[18] S. C. H. Leung, Z. Zhang, D. Zhang, X. Hua, and M. K. Lim, "A meta-heuristic algorithm for heterogeneous fleet vehicle routing problems with two-dimensional loading constraints," *European Journal of Operational Research*, vol. 225, no. 2, pp. 199–210, 2013.

[19] O. Dominguez, A. A. Juan, B. Barrios, J. Faulin, and A. Agustin, "Using biased randomization for solving the two-dimensional loading vehicle routing problem with heterogeneous fleet," *Annals of Operations Research*, vol. 236, no. 2, pp. 383–404, Jan 2016.

[20] D. Zhang, R. Dong, Y.-W. Si, F. Ye, and Q. Cai, "A hybrid swarm algorithm based on abc and ais for 2l-hfcvrp," *Applied Soft Computing*, vol. 64, pp. 468–479, 2018.

[21] N. R. Sabar, A. Bhaskar, E. Chung, A. Turky, and A. Song, "An adaptive memetic approach for heterogeneous vehicle routing problems with two-dimensional loading constraints," *Swarm and Evolutionary Computation*, vol. 58, Article ID 100730, 2020.

[22] R. Baldacci and A. Mingozzi, "A unified exact method for solving different classes of vehicle routing problems," *Mathematical Programming*, vol. 120, no. 2, pp. 347–380, 2009.

[23] L. Junqueira, *Modelos e Algoritmos para Problemas Integrados de Roteamento e Carregamento de Veículos*, PhD thesis, Universidade Federal de São Carlos, São Carlos, 2013.

[24] C. S. Chen, S. M. Lee, and Q. S. Shen, "An analytical model for the container loading problem," *European Journal of Operational Research*, vol. 80, no. 1, pp. 68–76, 1995.

[25] C. D. J. Waters, "A solution procedure for the vehicle-scheduling problem based on iterative route improvement," *Journal of the Operational Research Society*, vol. 38, no. 9, pp. 833–839, 1987.

[26] S. Lin, "Computer solutions of the traveling salesman problem," *Bell System Technical Journal*, vol. 44, no. 10, pp. 2245–2269, 1965.

[27] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[28] V. Černý, "Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm," *Journal of Optimization Theory and Applications*, vol. 45, no. 1, pp. 41–51, 1985.

[29] D. S. Johnson, C. R. Aragon, L. A. McGeoch, and C. Schevon, "Optimization by simulated annealing: an experimental evaluation; part i, graph partitioning," *Operations Research*, vol. 37, no. 6, pp. 865–892, 1989.

[30] D. S. Johnson, C. R. Aragon, L. A. McGeoch, and C. Schevon, "Optimization by simulated annealing: an experimental evaluation; part ii, graph coloring and number partitioning," *Operations Research*, vol. 39, no. 3, pp. 378–406, 1991.