

Article

A Matheuristic Approach to the Integration of Three-Dimensional Bin Packing Problem and Vehicle Routing Problem with Simultaneous Delivery and Pickup

Ana Moura ¹, Telmo Pinto ^{2,3,*}, Cláudio Alves ³ and José Valério de Carvalho ³

¹ University of Aveiro, GOVCOPP, Systems for Decision Support Research Group, Campus Universitário de Santiago, 3810-193 Aveiro, Portugal

² University of Coimbra, CEMMPRE, Department of Mechanical Engineering, 3000-370 Coimbra, Portugal

³ University of Minho, Centro ALGORITMI, Campus de Gualtar, 4710-057 Braga, Portugal

* Correspondence: telmo.pinto@uc.pt; Tel.: +351-239-790-717

Abstract: This work presents a hybrid approach to solve a distribution problem of a Portuguese company in the automotive industry. The objective is to determine the minimum cost for daily distribution operations, such as collecting and delivering goods to multiple suppliers. Additional constraints are explicitly considered, such as time windows and loading constraints due to the limited capacity of the fleet in terms of weight and volume. An exhaustive review of the state of the art was conducted, presenting different typology schemes from the literature for the pickup and delivery problems in the distribution field. Two mathematical models were integrated within a matheuristic approach. One model reflects the combination of the Vehicle Routing Problem with Simultaneous Delivery and Pickup with the Capacitated Vehicle Routing Problem with Time Windows. The second one aims to pack all the items to be delivered onto the pallets, reflecting a three-dimensional single bin size Bin Packing Problem. Both formulations proposed—a commodity-flow model and a formulation of the Three-Dimensional Packing Problem must be solved within the matheuristic. All the approaches were tested using real instances from data provided by the company. Additional computational experiments using benchmark instances were also performed.

Keywords: vehicle routing problem with simultaneous delivery and pickup; capacitated vehicle routing problem with time windows; three-dimensional bin packing problem

MSC: 90C11; 90C27; 90C59



Citation: Moura, A.; Pinto, T.; Alves, C.; Valério de Carvalho, J. A Matheuristic Approach to the Integration of Three-Dimensional Bin Packing Problem and Vehicle Routing Problem with Simultaneous Delivery and Pickup. *Mathematics* **2023**, *11*, 713. <https://doi.org/10.3390/math11030713>

Academic Editor: Shih-Wei Lin

Received: 30 December 2022

Revised: 23 January 2023

Accepted: 27 January 2023

Published: 31 January 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

1.1. Pickup and Delivery Problems

The pickup and delivery problem is commonly faced in real-world logistics operations. Accordingly, the number of contributions in this area has been increasing in the literature. Due to the extensive number of approaches and denominations in this field, in [1] suggested a comprehensive classification for the static pickup and delivery problems. The static denomination is due to the fact that all information about the problem is deterministic and known beforehand. In this classification, the scheme $[x|y|z]$ is used, where the x field defines the relation between the origins and the destinations:

many-to-many (M-M):	if any customer or the depot can be the origin or the destination of goods;
one-to-many-to-one (1-M-1):	if the source of delivered goods and the destination of picked goods is only the depot;
one-to-one (1-1):	if each good has to be transported from a single origin to a single destination.

The y field defines how the loading operations must be performed. If one customer is visited exactly once and loading and unloading operations must be performed simultaneously, the authors used the denomination “PD”. On the contrary, if this requirement does not apply, these operations can be combined or not (“P-D”). If it is only possible to deliver or collect at each customer, then the notation “P/D” is used instead. In the y field, problems with trans-shipments are defined by associating the letter “T”. The last field of this classification scheme, z , corresponds to the number of used vehicles. For the cases in which it is impossible to represent a field, the authors use the notation “-”. As in vehicle routing problems, in some pickup and delivery contexts, the information is dynamic, meaning that it is not known in advance or it is updated when performing the routes. These problems are known as dynamic pickup and delivery problems. A comprehensive survey on these problems is presented in [2].

Another comprehensive classification is proposed in [3,4]. According to these authors, the general pickup and delivery problems can be divided into two classes. In the first class, the goods are delivered or collected to/from one or more depots. These problems are usually named as Vehicle Routing Problems with Backhauls. On the other hand, in the second class, there is a flow of items from pickup customers to delivery customers. Problems belonging to this second class are usually known as Vehicle Routing Problems with Pickups and Deliveries (VRPPD).

1.2. Pickup and Delivery Problems with Loading

Other routing problems deal with different packing constraints when compared with those presented for the CVRPTW with loading constraints [5–7]. In some of those problems, routing aspects are similar but the vehicle is replaced by a ship or the cargo is palletized and, consequently, the pallets are items to be loaded in the vehicles. In other situations, items may not be in contact and thus they cannot be transported in the same compartment or even in the same vehicle. However, other problems must combine loading constraints with pickup and delivery.

Palletized items arise in many situations, such as the distribution of chip-boards. It is considered that the demand of each customer is composed of different types of chip-boards. The chip-boards of the same type for the same customer are jointly palletized, performing an item. The items are delivered to customers using a fleet usually composed of the same type of vehicles, each divided into a given number of piles (stacks). However, long chip-boards may occupy more than one pile, and thus different chip-board types may require a different number of occupied piles. Additionally, sequential constraints are imposed. Consequently, it is clear that some holes may arise in the layout, which may cause load instability. In these cases, such holes may be filled by bulk material, and thus supporting constraints can be ignored.

The described problem is denominated by the One Vehicle Loading Problem (1-VLP). The first work addressing the 1-VLP is presented in [8]. The authors proposed a tabu search and an ant colony optimization algorithm. A variable neighborhood search approach that provided an upper bound and possible valid cuts for a branch-and-cut algorithm is proposed in [9]. Both works referred to above were tested in real instances by an Austrian timber distribution company.

Another problem considering palletized items is presented in [10]. The demand of each customer is composed of three-dimensional items previously assigned to a given pallet. It is considered that one pallet is sufficient to carry the total demand of a given customer. The pallets are then loaded into the vehicle. As stated by the authors, this problem corresponds to solving, for each customer, an instance of CVRPTW with three-dimensional loading with supporting constraints and variable orientation of items. Sequential constraints are not considered since the vehicle driver can reach the pallets.

Some routing problems deal with situations in which it is common to transport goods that are incompatible in the sense that they may not share the same space in the vehicle. In such cases, items may be transported in the same vehicle but in different compartments.

The corresponding problem is known as the Multi-Compartment Vehicle Routing Problem (MC-VRP). The main difference between MC-VRP and the 1-VLP referred to above is that items belonging to different compartments are not in contact, while in the 1-VLP one item may occupy more than a single pile. This feature may be suitable for transporting conflicting or strongly heterogeneous items within the same vehicle. Some real applications for this problem can be found in the fuel distribution, when a single vehicle transports different types of petrol products [11–14]. Additionally, legal requirements must impose that items must be transported in different compartments. One example of this situation is the transportation of provisions to farms where, despite the separation in different compartments, the food for a given specie must be loaded in the same compartment in future transportation, as referred to in [15].

Other situations arise when items need different transport conditions, as it happens when serving convenience stores with dry, refrigerated, and frozen items within the same vehicle [16]. A similar problem is proposed in [17]. However, items may not be transported within the same vehicle and thus multiple trips must be performed. The authors denominated this problem as minimum multiple trip vehicle routing problem. Time windows for different type of items are also considered.

In other situations, it is necessary to ensure the transportation from pickup vertices to delivery vertices. For instance, the Pickup and Delivery Traveling Salesman Problem with Loading constraints (PDT SPL) considers the assumption that pickups and deliveries are performed according to a LIFO or a FIFO policy. Since only one dimension is considered, the vehicle can be seen as a single stack with items placed on its top. Consequently, LIFO constraints are only satisfied if, when serving one customer, the item for this customer can be unloaded in a straight movement, i.e., no items are between the item to be unloaded and the rear side of the vehicle. On the other hand, FIFO constraints impose that items are unloaded by the sequence in which they were loaded.

Some works extended the PDT SPL with LIFO constraints to the case where two routes are considered: one to collect items from pickup customers and another to deliver them to delivery customers. Additionally, items are placed on multiple stacks. These problems are known as Double Traveling Salesman with Multiple Stacks (DTSMS). A survey of PDT SL and DTSMS can be found in [18].

Pickup and delivery problems considering two- or three-dimensional items have received increasing attention in the literature. Considering the two-dimensional case, a constraint programming model approach is suggested in [19] considering transportation between nodes in the same graph. Considering the transportation from and to the depot, an insertion heuristic for the mixed linehauls and backhauls is addressed in [20] where LIFO constraints are explicitly addressed. In [21], three different variable neighborhood search (VNS)-based algorithms are designed, showing that general and skewed variants of VNS can lead to good solutions in a reasonable amount of time. In [22], three-dimensional items are considered within a local search method. The three-dimensional case is addressed in [23], addressing clustered backhauls, which assumes that backhaul customers can only be visited after serving the linehaul ones within the same route.

This paper addresses a real pickup and delivery problem arising in a Portuguese automotive company. The problem involves a set of practical considerations that have received little attention in the vehicle routing literature. These considerations are related to the optimization of the cargo that must be delivered and collected.

The first problem addressed in this work is an extension of Capacitated Vehicle Routing Problem with Time Windows (CVRPTW) where each node of the graph is associated with two quantities representing the commodities to be delivered and collected, called Vehicle Routing Problem with Simultaneous Delivery and Pickup and Time Windows (VRPSDPTW). Additionally, we address a second problem to minimize the number of pallets needed for each supplier. According to the Wäscher et al. [24] typology, it corresponds to a three-dimensional single bin size Bin Packing Problem, hereafter called 3D-BPP. It is worth noting that the contribution of the 3D-BPP approach in this scenario is twofold since

it can also be applied while loading pallets into the vehicles aiming to maximize vehicle usage.

This paper is organized as follows. In Section 2, the real case study is presented, and all the operational constraints are stated. In Section 3, two mixed integer programming model formulations for two different combinatorial optimization problems are presented and explained in detail. The way these two different models are integrated using a hybrid approach is given in Section 4 while the results from computational experiments are described in Section 5. Finally, some conclusions are drawn in Section 6.

2. Problem Description

The problem tackled in this work is a real problem of a Portuguese automotive production company belonging to an international corporation. The vast majority of the supplied components (roughly 86%) come from Southeast Asia in the form of Complete Knock Down (CKD) kits which are stored in a specific part of the warehouse. A set of subcontracted automotive suppliers provides the remaining amount of supplied components. However, a subgroup of parts from the CKD kits is not directly incorporated into the vehicle. These parts are transported and assembled at a subset of the subcontracted suppliers referred to above and then introduced in the production process at the company. Therefore, for a small group of suppliers, the flow of components is conducted in both directions.

It is assumed that the company uses its homogeneous fleet to ensure the distribution and collection process between the company and its suppliers. Each vehicle leaves the company with components to the suppliers, arriving within a given time window. This happens since the company has a predefined time to make all deliveries and collections. The vehicle must remain at the supplier during a given service time needed both to unload the demands and to collect the subcontracted components to carry them back to the company (hereafter, also known as a depot).

The demand at each supplier is composed of a set of boxes, and all the cargo must be palletized. Then it is necessary to pack all the boxes on pallets aiming to maximize the volume utilization of each one. This process leads to a minimum number of pallets used. Considering that each supplier has a given demand and could receive more than one pallet, the main objective is to minimize the number of needed pallets for each supplier.

The amount of delivered and collected cargo varies according to the supplier since subcontracted components are not uniform. However, the load to be picked up has a smaller volume than (or at least equal to) the delivered cargo. As stated above, all the load is palletized, but it is possible to pile more than one pallet if the height of the pallets allows it. Additionally, some operational constraints related to routing and packing must be satisfied. Some of those constraints, namely the time windows and the items to be collected or delivered, directly impact the suppliers visiting sequence. The constraints can be grouped into two sets:

- G1 : Routing constraints: time windows constraints, delivered, and picked-up demands, capacity constraints in terms of weight/number of pallets);
- G2 : Loading constraints (capacity constraints in terms of volume; box orientation constraints—for each box, a single dimension is allowed to act as the height of the box; positioning constraints—all the cargo must be completely inside the pallet, and each box should not be overlapped).

The global objective is to determine the set of routes that minimizes the costs of the distribution and collection operations of the components while satisfying all the demanded/supplied quantities at each entity and all the operational constraints. Hereafter, customers and suppliers will be referred to indistinctly to be able to use terminology in the context of routing problems.

3. Mixed-Integer Programming Models

Two mixed-integer programming models are proposed to be used in a complementary way by the hybrid algorithm. The first one formulates the VRPSDPTW, which minimizes

the total distance traveled. The second model is a formulation for the 3D-BPP, which minimizes the number of pallets needed for each supplier. The VRPSDPTW and the 3D-BPP are NP-hard in the strong sense. This fact and the necessity to solve real-world instances have pushed researchers to develop heuristic algorithms. In contrast, and mainly in the packing problems, exact algorithms have received little attention in the literature. When this problem deals with a small number of suppliers and cargo, these two problems could be solved using its mathematical formulations. The VRPSDPTW model developed in Section 3.1 is a two-index commodity-flow formulation. The 3D-BPP model defines the loading pattern of the boxes onto pallets that will be loaded in the vehicle. Those two formulations are based on [25,26].

3.1. A Two-Index Commodity-Flow Model for the VRPSDPTW

This formulation describes the motion of the vehicle and filling level in the road network and determines the amount of the loads that flow along an arc.

The VRPSDPTW can be defined in a directed graph $G = (C, A)$ where the set of nodes $C = \{1, \dots, |C|\}$ represents the depot ("1") and the set of geographically dispersed suppliers, and $A = \{(i, j) : i, j \in C, i \neq j\}$ corresponds to the set of arcs in G . The length of each arc d_{ij} with $(i, j) \in A$ corresponds to the distance between nodes i and j . Each supplier $i \in C \setminus \{1\}$ has a set of boxes (d_i) that must be delivered and a set of boxes that must be picked up (p_i) from/to the depot using a homogeneous fleet $V = \{1, \dots, NV\}$ with a maximum weight capacity Q . Each supplier has an associated service time st_i to perform the loading and unloading operations and a given time window $[a_i, b_i]$, in which the vehicle must arrive. The depot also has a time window $[a_1, b_1]$ that will define the starting and the maximum time available for the distribution and collection process.

This model considers two-index binary decision variables x_{ij} taking value 1 if the arc $(i, j) \in A$ is traversed by a vehicle, and 0 otherwise. To ensure time constraints, the set of decision variables s_i represents the arrival instant of time at each node $i \in C$. These variables are commonly used in vehicle routing models. However, and since this model considers simultaneous deliveries and pickups, new continuous variables are employed to represent the amount of load (to deliver or to collect) that flows along each arc, as follows:

- α_{ij} : the amount of delivery load carried along an arc $(i, j) \in A$;
- μ_{ij} : the amount of picked up load carried along an arc $(i, j) \in A$.

The objective function corresponds to the minimization of the total distance traveled (Equation (1)):

$$\min \sum_{i=1}^{|C|} \sum_{j=1}^{|C|} d_{ij} \times x_{ij} \tag{1}$$

subject to:

$$\sum_{i=1, i \neq j}^{|C|} x_{ij} = 1, \forall j \neq 1 \in C \tag{2}$$

$$\sum_{j=1, i \neq j}^{|C|} x_{ij} = 1, \forall i \neq 1 \in C \tag{3}$$

$$\sum_{i=1}^{|C|} x_{1i} = NV \tag{4}$$

$$s_i + st_i + d_{ij} \leq s_j + M \times (1 - x_{ij}), \forall i, j \in C \tag{5}$$

$$a_i \leq s_i \leq b_i, \forall i \in C \tag{6}$$

$$\sum_{i=1, i \neq j}^{|C|} \alpha_{ij} - \sum_{i=1, i \neq j}^{|C|} \alpha_{ji} = d_j, \forall j \neq 1 \in C \tag{7}$$

$$\sum_{i=1, i \neq j}^{|C|} \mu_{ji} - \sum_{i=1, i \neq j}^{|C|} \mu_{ij} = d_j, \forall j \neq 1 \in C \tag{8}$$

$$\alpha_{ij} + \mu_{ij} \leq M^1 \times x_{ij}, \forall i, j : j \neq i \in C \tag{9}$$

$$0 \leq \alpha_{ij} \leq Q, \forall i, j : j \neq i \in C \tag{10}$$

$$0 \leq \mu_{ij} \leq Q, \forall i, j : j \neq i \in C \tag{11}$$

$$\sum_{i=1}^{|C|} \alpha_{ij} \geq d_j, \forall j \neq 1 \in C \tag{12}$$

$$\sum_{j=1}^{|C|} \mu_{ij} \geq p_i, \forall i \neq 1 \in C \tag{13}$$

$$x_{ij} \in \{0, 1\}, \forall (i, j) \in A; \tag{14}$$

$$s_i \geq 0, \forall i \in C; \tag{15}$$

$$\mu_{ij} \geq 0, \forall (i, j) \in A, i \neq 1; \tag{16}$$

$$\alpha_{ij} \geq 0, \forall (i, j) \in A, j \neq 1; \tag{17}$$

Constraints (2) and (3) ensure that each supplier is visited exactly once by a vehicle, and (4) states that no more routes than the number of available vehicles are created. Inequalities (5) and (6) are related to the time windows constraints (M is a very large number).

The first establishes the relation between arrival times at clients i and j when the arc (i, j) is used, and the second guarantees that the vehicle arrives within the related time window. Additionally, (5) forces a specific order for suppliers visiting within the routes, which ensures that no sub-tours are generated without the depot. Constraints (7) and (8) are the balance equations to satisfy the delivery and the pickup demands of a customer, respectively. The consideration of those equations also eliminates sub-tours occurrences. Constraints (9)–(11) ensure that the capacity of the vehicle is not exceeded and the “big- M value” of the disjunctive constraint (9) is set to $M^1 = Q$. Constraints (12) and (13) are bounds of the total quantities of boxes entering and leaving a supplier’s location. The consideration of these two constraints improves the solving process. The last Constraints (14)–(17) define the domain of the decision variables.

3.2. A 3D-BPP Model

This model is used to pack all the boxes for each customer on pallets before the vehicle leaves the depot. Therefore, and for each customer, there is a set of boxes for each customer ($B = \{1, \dots, |B|\}$), where each box is characterized by its original length, width, and height dimensions Lbx_b, Wbx_b, Hbx_b with $b \in B$, and allowed orientations for the height dimension, i.e., on the axis perpendicular to the base of the pallet (OzL_b, OzW_b, OzH_b). Several standard pallets are available: let V be the set of all pallets $V = \{1, \dots, |V|\}$ and each one is also characterized by their physical dimensions Xv_k, Yv_k, Zv_k . A cost assigned

to the use of each pallet C_k is also considered. To pack all the available boxes and identify their final positioning on the pallets, several variables are defined as follows:

- D_k is a binary variable that indicates if pallet k is used or not.
- β_{kb} is a binary variable that indicates if box b is on pallet k or not.
- Three integer variables XX_b, YY_b, ZZ_b give the pallet us dimensions.
- Three variables related to the centroid of gravity center coordinates of each box x_b, y_b, z_b .
- The set of variables, Xbx_b, Ybx_b, Zbx_b is related to the chosen dimension of each box to be placed on x, y, z -axis of the pallet.
- For each axis x, y and z there is also a set of binary variables that allow for the rotation of each box: $bxL_b, bxW_b, bxH_b; byL_b, byW_b, byH_b; bzL_b, bzW_b, bzH_b$, respectively.
- For the non-overlapping constraints and each axis, there are several binary variables: P_{bl}, Q_{bl} for x -dimension; R_{bl}, S_{bl} for y -dimension; and T_{bl}, U_{bl} for z -dimension.
- The last binary variable, G_{bl} guarantees that the boxes b and l are loaded onto the same pallet.

The objective is to load all the boxes of each supplier onto a minimum number of pallets, aiming to minimize the number of vehicles used. Therefore, the objective function of the 3D-BPP is defined as the minimization of the number of pallets needed to pack all the demands of each supplier (or the costs associated with the use of pallets) as follows in (18):

$$\min \sum_{k=1}^{|V|} C_k \times D_k \tag{18}$$

Subject to:

$$x_b \geq \frac{Xbx_b}{2} \wedge y_b \geq \frac{Ybx_b}{2} \wedge z_b \geq \frac{Zbx_b}{2}; \forall b \in B \tag{19}$$

$$x_b \leq XX_b - \frac{Xbx_b}{2} \wedge y_b \leq YY_b - \frac{Ybx_b}{2} \wedge z_b \leq ZZ_b - \frac{Zbx_b}{2}; \forall b \in B \tag{20}$$

$$XX_b = \sum_{k=1}^v Xv_k \times \beta_{kb} \wedge YY_b = \sum_{k=1}^v Yv_k \times \beta_{kb} \wedge ZZ_b = \sum_{k=1}^v Zv_k \times \beta_{kb}; \forall b \in B \tag{21}$$

$$\sum_{k=1}^v \beta_{kb} = 1; \forall b \in B \tag{22}$$

$$\sum_{b=1}^{|B|} \beta_{kb} \leq bx \times D_k; \forall k \in V \tag{23}$$

$$\sum_{b=1}^{|B|} (Lbx_b \times Wbx_b \times Hbx_b) \beta_{kb} \leq (Xv_k \times Yv_k \times Zv_k) D_k; \forall k \in V \tag{24}$$

This first set of constraints is related to the positioning and volume constraints. To simplify the positioning of each box on a pallet, each box is characterized by its centroid of gravity center given by its x, y , and z coordinates. Constraints (19)–(21) combined with Constraints (22) guarantee that each box is positioned on a single pallet. Constraints (23) defines whether the pallet is used or not, and constraints (24) is used to prevent over-packing attempts.

$$\frac{x_b + x_l}{2} \leq (x_b - x_l) + M^x \times (P_{bl} + Q_{bl}) \wedge \frac{x_b + x_l}{2} \leq (x_b - x_l) + M^x \times (P_{bl} + Q_{bl} + 1) ; \forall b, l : l > b \in B \tag{25}$$

$$\frac{y_b + y_l}{2} \leq (y_b - y_l) + M^y \times (R_{bl} + S_{bl}) \wedge \frac{y_b + y_l}{2} \leq (y_b - y_l) + M^y \times (R_{bl} + S_{bl} + 1) ; \forall b, l : l > b \in B \tag{26}$$

$$\begin{aligned} \frac{z_b+z_l}{2} &\leq (z_b - z_l) + M^z \times (T_{bl} + U_{bl}) \wedge \\ \frac{z_b+z_l}{2} &\leq (z_b - z_l) + M^z \times (T_{bl} + U_{bl} + 1) \end{aligned} ; \forall b, l : l > b \in B \tag{27}$$

$$P_{bl} + R_{bl} + T_{bl} \leq 3; \forall b, l : l > b \in B \tag{28}$$

$$\beta_{bk} - \beta_{kl} \leq G_{bl} + 1; \forall b, l : l > b \in B \tag{29}$$

Non-overlapping constraints are used to prevent that different boxes from occupying the same physical location on a pallet (from constraints (25) to (29)). Non-overlapping is guaranteed if at least one of sets of the Big-M constraints (25), (26), or (27) is active. M^x , M^y , and M^z are adequate upper bounds computed according to the dimensions of the box set.

$$\begin{aligned} Xbx_b &= bxL_b \times Lbx_b + bxW_b \times Wbx_b + bxH_b \times Hbx_b \wedge \\ bxL_b + bxW_b + bxH_b &= 1; \end{aligned} \forall b \in B \tag{30}$$

$$\begin{aligned} Ybx_b &= byL_b \times Lbx_b + byW_b \times Wbx_b + bzH_b \times Hbx_b \wedge \\ byL_b + byW_b + byH_b &= 1; \end{aligned} \forall b \in B \tag{31}$$

$$\begin{aligned} Zbx_b &= bzL_b \times Lbx_b + bzW_b \times Wbx_b + bzH_b \times Hbx_b \wedge \\ OzL_b \times bzL_b + OzW_b \times bzW_b + OzH_b \times bzH_b &= 1; \end{aligned} \forall b \in B \tag{32}$$

$$\begin{aligned} bxL_b + byL_b + bzL_b &= 1 \wedge bxW_b + byW_b + bzW_b = 1 \wedge \\ bxH_b + byH_b + bzH_b &= 1 \end{aligned} \forall b \in B \tag{33}$$

Each box is characterized by its length, width, and height dimensions. To guarantee the boxes' height orientation constraints and maximize the freeloading space, when a box is loaded onto a pallet, each of these dimensions must be assigned to a given axis x, y , and z . Because of the orientation constraint in height, the second equation of the set of Constraints (32) considers the permitted dimension(s) to be placed along the z -axis. Once this dimension is chosen, the other two dimensions could be placed along the x and y -axis, only bearing in mind the best position to maximize the free space.

$$\begin{aligned} x_l &\leq \frac{XX_l}{2} + M^{xx} \sum_{b=1}^{l-1} G_{bl} \wedge y_l \leq \frac{YY_l}{2} + M^{yy} \sum_{b=1}^{l-1} G_{bl} \wedge \\ z_l &\leq \frac{ZZ_l}{2} + M^{zz} \sum_{b=1}^{l-1} G_{bl} \end{aligned} \forall l \in B \tag{34}$$

Due to the geometry of the three-dimensional packing problems, several symmetrical loading patterns may lead to multiple optimal solutions to the same problem, increasing the CPU time. As in [27], some symmetry-breaking constraints (34) are used to avoid the problem symmetry. The "Big-M values" (M_{xx} , M_{yy} , and M_{zz}) are upper bounds computed according to the pallet us dimensions in the x, y , and z -axis.

4. The Hybrid Approach for the VRPSDPTW

Both the VRPSDPTW and the 3D-BPP are NP-Hard optimization problems. This complexity can prevent achieving exact solutions in an acceptable time for large-scale instances. Therefore, we resort to hybrid strategies that use the proposed and innovative models with the fast solutions obtained by a GRASP algorithm within a matheuristic approach.

The general architecture of this Hybrid approach is summarized in the procedure presented in Algorithm 1. The first step is the palletization process, packing the demand of each customer. For each set of boxes of each customer, an instance of 3D-BPP is solved. After this process, the demand of each customer is not composed of a set of boxes but a set of pallets instead. Then a constructive phase is performed through a Greedy Randomized Algorithm (Algorithm 2) followed by the local search phase resorting to a local search procedure (Algorithm 3). This strategy encourages the search to examine unvisited regions of the solution space or generate solutions that significantly differ from those previously visited.

Algorithm 1: Hybrid Algorithm.**Input:**

I : Instance of the problem;
 C : set of costumers;
 B_i : set of boxes to be delivered to costumer i ;
 P_i : set of pallets to be delivered to costumer i ;
 P set of pallets;

Output:

S : set of solutions;

begin

```

 $S \leftarrow \{\}$ ;
for each  $i \in C$  do
   $P_i \leftarrow \text{3D-BPPModel}(B_i)$ ;
   $P \leftarrow P_i$ ;
end for
ConstructionPhase ( $C, P, S$ ) // Algorithm 2
   $S \leftarrow \text{GreedyRandomizedConstruction}(C, P)$ ;
  return  $S$ ;
LocalSearchPhase ( $S$ ) // Algorithm 3
   $S' \leftarrow \text{SwapProcedure}(S)$ ;
  return  $S$ ;

```

end**Algorithm 2:** GreedyRandomizedConstruction.**Input:**

C : the set of costumers;
 C_0 : the set of sorted costumers;
 Cl : the set of clusters;
 S : the set of solutions;
 P : the set of pallets;
 P : the set of solutions;

Output:

S : set of solutions;
 Cl : set of clusters;

begin

```

repeat
   $C_0 \leftarrow \text{SortingCostumers}(C)$ ;
   $RCL \leftarrow cl \in \{C_0 | d_{1i} \leq d^{max} - \alpha \times (d^{max} - d^{min})\}$ ;
   $cl \leftarrow \text{BuildCluster}(RCL)$ ;
   $s' \leftarrow \text{CommodityFlowModel}(cl)$ ;
   $s' \leftarrow \text{3D-BPPModel}(P, s')$ ;
  if ( $s'$  feasible) then
     $S \leftarrow S \cup s'$ ;
     $Cl \leftarrow Cl \cup cl$ ;
  end if
  else
     $C \leftarrow C \cup cl$ ;
  end if
until  $C \neq 0$ ;

```

end

Algorithm 3: Local search procedure.**Input:***Cl*: set of clusters*S*: set of solutions*P*: set of pallets;**Output:***S*: set of solutions**begin** **repeat** *cl'* pairs ← *RandomSelection(Cl)*; **for** (*i* = 1 **to** *Number of Swaps*) **do** *cl* ← *SwapCostumers(cl')*; *s'* ← *Commodity – flowModel(cl)*; **if** (*f(s') < f(S)*) **then** **if** (*3D-BPPModel(P, s')* returns a feasible packing) **then** | *S* ← *StoreSolution(s')*; **end if** **else** | *IgnoreSwap(cl)* **end if** **end if** **else** | *IgnoreSwap(cl)* **end if** **end for** **until** *Stop criteria is met*;**end**

More precisely, and as shown in Algorithm 2, the set of customers (*C*) is sorted according to the increasing order of distance to the depot through the process *SortingCustomers(C)* giving rise to the list *Co*. Hence, clusters of customers will be built, ensuring that the sum of the demand of customers (pallets) in each cluster does not exceed the capacity of the vehicle (in terms of the total volume of the pallets). This issue is important to avoid infeasible solutions in terms of packing. However, as it will be seen, it is not sufficient to guarantee a feasible solution. It is important to remark that one customer should be assigned to exactly one vehicle. This clustering process is a part of the constructive phase of the GRASP approach and uses a restricted candidate list (RCL). The elements of RCL will be randomly selected from the list *Co* using a threshold β as follows:

$$\beta = d^{max} - \alpha \times (d^{max} - d^{min}), \quad (35)$$

where

- d^{max} is the distance from the depot to the furthest customer;
- d^{min} is the distance from the depot to the closest customer;
- α is a parameter varying in the range $[0, 1]$.

Consequently, the RCL will have a different size according to the value of α . The size of RCL will resort to the size of *C* if $\alpha = 0$, i.e., all customers will be included in RCL. Conversely, if $\alpha = 1$, the RCL will have only one element: the customer closest to the depot. Since clusters are built by randomly selecting customers from the RCL (*BuildCluster(RCL)*), the use of Equation (35) will provide randomness to the clustering process. For instance, and taking into account only the extreme points of the range, the process is completely random if $\alpha = 0$ and totally greedy in the conversely ($\alpha = 1$). Subsequently, for each cluster *cl*, a route is obtained using the commodity-flow model presented in Section 3 (*CommodityFlowModel(cl)*);

ensuring constraints of type $G1$. Immediately after, the feasibility in terms of packing is verified to ensure the constraints of type $G2$ ($3D - BPPModel(P, s')$). This means that a feasible three-dimensional packing into one vehicle must be obtained for this cluster, and the related cargo is loaded into the vehicle, using the 3D-BPP model using the maximization of the vehicles' volume usage as the objective function.

Regardless of the clusters being built while ensuring the capacity of the vehicle in terms of the number of pallets, it is not ensured that the 3D-BPP model always returns a feasible solution. These situations arise due to the geometry and volume of the pallets, which can prevent a given constraint of the group $G2$ can be satisfied. Consequently, in such cases, the construction of the cluster is aborted, the customers are reassigned to the list C , new clusters are built, and the whole procedure is repeated. The algorithm stops when each customer is assigned to a given route and all clusters are feasible, i.e., satisfying all the constraints $G1$ and $G2$.

Therefore, the output of Algorithm 2 provides a feasible solution for the main problem, composed of a set of clusters of customers. Nevertheless, even if an optimal solution is achieved for each cluster, it is not possible to ensure that it is an optimal global solution for the problem, as it relies on partitioning the set of customers into different sets. Several different partitions can lead to significantly distinct solutions. As referred to above, and aiming to achieve improved solutions, a local search is applied immediately after using the obtained output of Algorithm 2 as input of the Algorithm 3.

Firstly, the algorithm randomly selects two clusters from the set Cl . For each pair, a given number of swaps is performed among two customers, each one from each cluster. The swap consists of exchanging a pair of customers from different clusters. The number of allowed swaps is equal to the number of customers in the smallest selected cluster, which gives several combinations of different clusters. Secondly, the commodity-flow model is applied to the resulting pairs of clusters, and the overall solution is assessed $Commodity - flowModel(cl)$. If this swap leads to any reduction in the objective function while satisfying the capacity of the vehicle, the feasibility of terms of loading (constraints of type $G2$) of both clusters is assessed using the 3D-BPP model proposed in Section 3. This model must be applied to load all the pallets of the related customers in each vehicle, with a different objective function: maximize the volume utilization. If a feasible packing is obtained for both clusters, then the solution is stored, considering the new clusters. On the contrary, if the swap procedure does not lead to any saving, or if there is an infeasible packing for at least one cluster, the swap is discarded. This overall procedure is repeated until a given stop criteria is met. Therefore, the best solution in a set of solutions S is returned if it corresponds to an improvement of the initial solution.

5. Computational Results

The global objective is to solve and optimize an operational distribution problem, including routing and packing issues, in a real-world scenario arising in this company. However, to evaluate the quality of the proposed approach, two different sets of computational experiments were conducted: one for real instances and another for benchmark instances. Real problem instances relied on data provided by the company. They were used to test not only the hybrid approach but also the mathematical models separately, as presented in Section 5.1. However, the set of suppliers of the company is relatively small, and the weight and volume of the boxes in each route are always smaller than the capacity of the vehicle. To test the efficiency of the presented approach and mathematical models in more challenging and large-size problem instances, computational experiments were conducted on some VRPSDPTW and 3D-BPP benchmark instances as shown in Section 5.2. The hybrid approach was implemented in C++, and the mathematical models were solved using the CPLEX solver. All the experiments were run on a desktop computer with Intel Core i7 vPro CPU at 2.2 GHz and 8-GB RAM. After preliminary experiments, α was experimentally set to 0.8.

5.1. Results for Real Instances of the VRPSDPTW and 3D-BPP (Routing and Loading)

According to the nature of the real data given by the company, two types of instances were created for 10 suppliers and one vehicle: one with a symmetric travel time matrix and another one with an asymmetric matrix. For each supplier, a given time window is provided (from 0 to 12 h) and the collected and delivered quantities are known (from 0 to 4 pallets). The collected quantities are always smaller than or equal to the delivered quantities. Those instances vary from 3 to 10 suppliers. The demand of each supplier varies from 1 to 4 different types of boxes and the average number of boxes for each supplier is 100. The pallets have standard dimensions (120 cm × 80 cm), and the maximum height of a pallet can vary between 80, 120, or 180 cm depending on the requirements of the logistics suppliers or on the product types. Six instances for the overall problem were solved using the commodity-flow model and the 3D-BPP model, for the symmetric and asymmetric instances, giving rise to twelve instances. Tables 1 and 2 present the obtained results in terms of CPU time in seconds (CPU), the total travel time in hours (Travel Time), the number of pallets (No. Pallets), and the optimality gap in percentage (Gap). Note that the CPU time for the 3D-BPP is defined in seconds for each instance.

Table 1. Experimental results for real and asymmetric instances.

Instances	Commodity-Flow			3D-BPP		
	Travel Time (h)	CPU (s)	Gap (%)	No. Pallets	CPU (s)	Gap (%)
1	2.46	1.45	0.00	31	0.05	0.00
2	2.95	1.65	0.00	34	0.03	0.00
3	2.46	1.90	0.00	25	0.01	0.00
4	2.46	1.00	0.00	34	0.04	0.00
5	2.95	1.61	0.00	30	0.03	0.00
6	2.46	2.34	0.00	30	0.03	0.00

Table 2. Experimental results for real and symmetric instances.

Instances	Commodity-Flow			3D-BPP		
	Travel Time (h)	CPU (s)	Gap (%)	No. Pallets	CPU (s)	Gap (%)
1	3.37	1.40	0.00	34	0.06	0.00
2	2.94	1.62	0.00	24	0.01	0.00
3	3.37	1.59	0.00	34	0.05	0.00
4	3.37	2.34	0.00	34	0.02	0.00
5	2.94	1.15	0.00	34	0.04	0.00
6	3.37	1.40	0.00	34	0.06	0.00

An optimal solution was always found for the whole set of real instances for each mathematical model. The computational time for each of the models is tiny. Furthermore, there are no significant differences between symmetric and asymmetric instances. It is worth noting that the 3D-BPP model is underused whether it only provides pallets of a small fraction of heterogeneity for each customer, even in real instances. For this reason, and to assess its efficiency, more computational experiments are presented in the sequel.

5.2. Experimental Results for Adapted Benchmark Instances

In order to assess the quality of the overall approach, more challenging instances than the ones presented in Section 5.1 are required. Therefore, we resort to two different sets of experiments:

- E1 set of tests to assess the efficiency of the 3D-BPP using benchmark provided in [28] (denoted in the literature by *Thpack9*, which is a set of 47 instances ranging from 2 to 5 item types and from 47 to 180 items);

E2 set of tests to assess both the VRPSDPTW and 3D-BPP models separately, as well as the hybrid approach in an integrated way, using benchmark instances provided in [28,29].

Concerning E1, the same set of instances was also used in [30]. These instances were used in contributions considering multiple containers whose objective is to minimize the number of containers needed to load the whole set of heterogeneous items. Therefore, all 47 instances were tested and presented in Table 3 as well as the minimum number of required containers (No. Cont.) and the CPU time (CPU (s)). The proposed model proved to be very efficient since it always achieves the optimal solution in very short computational times for all the instances. The minimum CPU time is 0.01 s (reaching 4 containers) and the maximum CPU time is 3.28 s (for 26 containers). However, it is not possible to establish an association between CPU time and the minimum number of containers: the time difference is negligible.

Table 3. Experimental results for the adapted benchmark instances for the 3D-BPP (pallets loading).

Instances	3D-BPP		Instances	3D-BPP	
	No. Cont.	CPU(s)		No. Cont.	CPU (s)
1	19	0.51	25	5	0.10
2	7	0.39	26	3	0.07
3	19	2.53	27	5	0.09
4	26	3.28	28	9	0.29
5	46	2.73	29	15	0.87
6	10	1.98	30	18	0.99
7	16	1.78	31	11	0.23
8	5	0.55	32	4	0.01
9	17	0.33	33	5	0.04
10	37	1.52	34	8	0.23
11	14	0.86	35	3	0.10
12	45	2.30	36	11	0.54
13	20	0.06	37	12	0.64
14	28	1.19	38	25	0.86
15	11	0.03	39	12	0.59
16	21	0.33	40	8	0.21
17	7	0.02	41	14	0.23
18	3	0.11	42	4	0.70
19	3	0.12	43	3	0.07
20	5	0.82	44	3	0.10
21	17	0.97	45	3	0.50
22	8	0.22	46	2	0.09
23	17	1.02	47	3	0.19
24	5	0.13			

Concerning the E2 experiments, and since the hybrid approach deal with routing and loading components, we resort to a set of instances provided in [29] combined with those from [28], performing a set of 21 instances. Only the first 20 customers were considered to adapt those instances to a single VRPSDPTW. The instances are symmetric and grouped in clustered suppliers (IC type), uniformly distributed suppliers (IR type), and a mix of IC and IR types (IRC types).

The computational results for this set are presented in Table 4, grouped by the assessed method: the commodity-flow model itself and the Hybrid approach. The minimum number of required vehicles (No. Veh), the CPU time (CPU (s)) and the optimality gap (GAP) are also presented.

An optimal solution is always obtained except for the last two instances IRC203 and IRC204. These instances correspond to the mix of clustered customers with Uniformly

distributed customers (mix IRC). In this particular case, after 2 h, the best bound is returned and used as an upper bound for comparison with the results of the hybrid approach. It is also possible to verify that IC instances have, on average, lower computational times than IR problems.

Comparing these results with the results obtained by the hybrid approach, optimal solutions are obtained in some instances (presented in bold). In the instances that only two vehicles are required in the best solution for the commodity-flow model (e.g., IR204, IR207, IR208, and IR211) is the 3D-BPP model solution of three vehicles, the result obtained by the hybrid approach is also three. In these cases, and since the hybrid approach solves both problems simultaneously, it is assumed that those solutions will be the optimal solution. For the remaining instances, good solutions were achieved in a very reduced computational time (roughly one second in the vast majority of cases).

Table 4. Experimental results for the adapted benchmark instances for the VRPSDPTW and 3D-BPP (routing and loading several vehicles).

Instances	Commodity-Flow			Hybrid GRASP		
	No. Veh/Dist	CPU (s)	GAP (%)	No. Veh/Dist	CPU (s)	GAP (%)
IC201/Thpack9_44	3/592	30.01	0	3/592	<1	0
IC202/Thpack9_44	3/592	33.07	0	3/592	<1	0
IC203/Thpack9_44	3/586	58.60	0	3/595	1.03	1.54
IC204/Thpack9_44	3/591	39.69	0	3/591	1.02	0
IC205/Thpack9_44	3/589	31.17	0	3/631	<1	7.13
IC206/Thpack9_44	3/588	33.29	0	3/609	<1	3.57
IC207/Thpack9_44	3/588	32.06	0	3/588	<1	0
IC208/Thpack9_44	3/588	32.68	0	3/598	<1	1.70
IR202/Thpack9_44	3/1198	196.36	0	3/1214	1.78	1.34
IR203/Thpack9_44	3/949	1764.03	0	3/949	1.83	0
IR204/Thpack9_44	2/849	296.10	0	3/1132	<1	-
IR205/Thpack9_44	3/1054	132.42	0	3/1069	<1	1.42
IR206/Thpack9_44	3/932	161.03	0	3/949	<1	1.82
IR207/Thpack9_44	2/903	905.81	0	3/1204	1.35	-
IR208/Thpack9_44	2/735	132.01	0	3/980	<1	-
IR209/Thpack9_44	3/931	233.23	0	3/945	<1	1.50
IR210/Thpack9_44	3/964	233.96	0	3/972	<1	0.83
IR211/Thpack9_44	2/912	429.98	0	3/1215	<1	-
IRC202/Thpack9_44	3/1374	1940.44	0	3/1374	2.09	0
IRC203/Thpack9_44	3/1089	>2 h	10.91	3/1205	3.10	10.65
IRC204/Thpack9_44	3/819	>2 h	5.36	3/899	3.68	9.77

The modeling approaches presented for this real case are helpful in the decision process. The company could solve a real problem in an acceptable time. Similar results were obtained even when the number of customers rose to 20. Given the very particular nature of this real problem, there is no approach in the literature to solve this specific problem. For this reason, the proposed methods address the special issues of this real problem. Despite that, the proposed models can be easily adapted to incorporate more constraints or address extended problems and/or their variants. Another advantage of the presented mathematical programming models is their adaptation to matheuristic-based approaches. Exact and heuristic approaches can be found in the literature for similar problems [19–23] but heuristics based on mathematical models, as discussed in this work, are still scarce. Despite the advantages presented, these approaches also have some limitations: one of them is the capacity to achieve good-quality solutions for large-size instances using only mathematical programming models. Another limitation is related to the used capacity of the vehicle. It would be interesting to address the rearrangement of items inside the

vehicle after each pickup. However, this inclusion would lead to an exponential increase in computing time, so it will always be necessary to conduct a trade-off analysis between the savings generated and obtaining solutions in an acceptable amount of time.

6. Conclusions

In this work, a real problem of an automotive production company belonging to an international corporation is presented. An exhaustive literature review was performed, considering the classification schemes in this field. Two mixed-integer programming models are proposed to address the problem. The first is a commodity flow model to address the Vehicle Routing Problem with Simultaneous Delivery and Pickup and Time Windows. The second model is the formulation for the 3D-BPP, which can be used in two different scenarios: to minimize the number of pallets needed for each supplier or to maximize the vehicle usage. Due to promising results and the nature of the two mathematical formulations, one of the main contributions of this work is the possibility of expand them to follow the growth of production of this company while maintaining good-quality solutions. For instance, it will be possible to address possible extensions or variants that the company may face in the future (e.g., heterogeneous fleet or mixed backhauls). Although the approach was used for real instances, adapted instances from the literature were also used to test the overall efficiency of the proposed methods. Most of the real instances were solved up to optimality in a very reduced computational time. For the remaining ones, good upper bounds were derived. For the adapted instances, the results seem promising since optimal solutions are obtained in some cases, while in the remaining cases, good-quality solutions can be derived.

Author Contributions: Conceptualization, A.M.; Methodology, A.M.; Software, A.M.; Validation, C.A. and J.V.d.C.; Formal analysis, A.M. and T.P.; Investigation, A.M., T.P., C.A. and J.V.d.C.; Data curation, A.M.; Writing—original draft, A.M. and T.P.; Writing—review and editing, T.P., C.A. and J.V.d.C.; Visualization, T.P.; Supervision, C.A. and J.V.d.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by national funds through FCT—Fundação para a Ciência e a Tecnologia, under the projects UIDB/00285/2020, UIDB/00319/2020. This work was supported by the Research Unit on Governance, Competitiveness and Public Policies (UIDB/04058/2020) + (UIDP/04058/2020), funded by national funds through the Foundation for Science and Technology, IP. This work was also funded by FEDER in the frame of COMPETE 2020 under the project POCI-01-0247-FEDER-072638.

Data Availability Statement: The data are not publicly available due to privacy or ethical restrictions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Berbeglia, G.; Cordeau, J.F.; Gribkovskaia, I.; Laporte, G. Static pickup and delivery problems: A classification scheme and survey. *TOP* **2007**, *15*, 1–31. [[CrossRef](#)]
2. Berbeglia, G.; Cordeau, J.F.; Laporte, G. Dynamic pickup and delivery problems. *Eur. J. Oper. Res.* **2010**, *202*, 8–15. [[CrossRef](#)]
3. Parragh, S.; Doerner, K.; Hartl, R. A survey on pickup and delivery problems. Part I: Transportation between customers and depot. *J. Betriebswirtschaft* **2008**, *58*, 21–51. [[CrossRef](#)]
4. Parragh, S.; Doerner, K.; Hartl, R. A survey on pickup and delivery problems. Part II: Transportation between pickup and delivery locations. *J. Betriebswirtschaft* **2008**, *58*, 81–117. [[CrossRef](#)]
5. Iori, M.; Martello, S. Routing problems with loading constraints. *TOP* **2010**, *18*, 4–27. [[CrossRef](#)]
6. Pinto, T.; Alves, C.; Carvalho, J.V.d. A branch-and-price algorithm for the vehicle routing problem with 2-dimensional loading constraints. In Proceedings of the International Conference on Computational Logistics, Lisbon, Portugal, 7–9 September 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 321–336. [[CrossRef](#)]
7. Gendreau, M.; Iori, M.; Laporte, G.; Martello, S. A Tabu Search Algorithm for a Routing and Container Loading Problem. *Transp. Sci.* **2006**, *40*, 342–350. [[CrossRef](#)]
8. Doerner, K.; Fuellerer, G.; Hartl, R.; Gronalt, M.; Iori, M. Metaheuristics for the vehicle routing problem with loading constraints. *Networks* **2007**, *49*, 294–307. [[CrossRef](#)]

9. Tricoire, F.; Doerner, K.; Hartl, R.; Iori, M. Heuristic and exact algorithms for the multi-pile vehicle routing problem. *OR Spectr.* **2011**, *33*, 931–959. [[CrossRef](#)]
10. Zachariadis, E.; Tarantilis, C.; Kiranoudis, C. The Pallet-Packing Vehicle Routing Problem. *Transp. Sci.* **2011**, *46*, 341–358. [[CrossRef](#)]
11. Cornillier, F.; Boctor, F.; Renaud, J. Heuristics for the multi-depot petrol station replenishment problem with time windows. *Eur. J. Oper. Res.* **2012**, *220*, 361–369. [[CrossRef](#)]
12. Cornillier, F.; Boctor, F.F.; Laporte, G.; Renaud, J. A heuristic for the multi-period petrol station replenishment problem. *Eur. J. Oper. Res.* **2008**, *191*, 295–305. [[CrossRef](#)]
13. Brown, G.; Graves, G. Real-Time Dispatch of Petroleum Tank Trucks. *Manag. Sci.* **1981**, *27*, 19–32. [[CrossRef](#)]
14. Pena, C.; Pinto, T.; Carvalho, M.S. A Two-Stage Heuristic for a Real Multi-compartment and Multi-trip Vehicle Routing Problem with Time Windows. In Proceedings of the Computational Science and Its Applications—ICCSA 2021, Cagliari, Italy, 13–16 September 2021; Springer International Publishing: Cham, Switzerland, 2021; pp. 274–289. [[CrossRef](#)]
15. Fallahi, A.; Prins, C.; Wolfler Calvo, R. A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. *Comput. Oper. Res.* **2008**, *35*, 1725–1741. [[CrossRef](#)]
16. Chajakis, E.; Guignard, M. Scheduling Deliveries in Vehicles with Multiple Compartments. *J. Glob. Optim.* **2003**, *26*, 43–78. [[CrossRef](#)]
17. Battarra, M.; Monaci, M.; Vigo, D. An adaptive guidance approach for the heuristic solution of a minimum multiple trip vehicle routing problem. *Comput. Oper. Res.* **2009**, *36*, 3041–3050. [[CrossRef](#)]
18. Pollaris, H.; Braekers, K.; Caris, A.; Janssens, G.; Limbourg, S. Vehicle routing problems with loading constraints: State-of-the-art and future directions. *OR Spectr.* **2014**, *37*, 297–330. [[CrossRef](#)]
19. Malapert, A.; Guéret, C.; Jussien, N.; Langevin, A.; Rousseau, L.M. Two-dimensional pickup and delivery routing problem with loading constraints. In Proceedings of the First CPAIOR Workshop on Bin Packing and Placement Constraints (BPPC'08), Paris, France, 22 May 2008.
20. Pinto, T.; Alves, C.; De, C.J.V.; Moura, A. An insertion heuristic for the capacitated vehicle routing problem with loading constraints and mixed linehauls and backhauls. *FME Trans.* **2015**, *43*, 311–318. [[CrossRef](#)]
21. Pinto, T.; Alves, C.; Valério de Carvalho, J. Variable neighborhood search algorithms for the vehicle routing problem with two-dimensional loading constraints and mixed linehauls and backhauls. *Int. Trans. Oper. Res.* **2020**, *27*, 549–572. [[CrossRef](#)]
22. Bartók, T.; Imreh, C. Pickup and Delivery Vehicle Routing with Multidimensional Loading Constraints. *Acta Cybern.* **2011**, *20*, 17–33. [[CrossRef](#)]
23. Bortfeldt, A.; Hahn, T.; Männel, D.; Mönch, L. Hybrid algorithms for the vehicle routing problem with clustered backhauls and 3D loading constraints. *Eur. J. Oper. Res.* **2015**, *243*, 82–96. [[CrossRef](#)]
24. Wäscher, G.; Haußner, H.; Schumann, H. An improved typology of cutting and packing problems. *Eur. J. Oper. Res.* **2007**, *183*, 1109–1130. [[CrossRef](#)]
25. Moura, A. Linear Programming applied to the Capacitated Simultaneous Delivery and Pick-up Problem. In Proceedings of the International Conference on Computational and Mathematical Methods in Science and Engineering, ICCMSE2018, Thessaloniki, Greece, 14–18 March 2018.
26. Moura, A. A model-based heuristic to the vehicle routing and loading problem. *Int. Trans. Oper. Res.* **2019**, *26*, 888–907. [[CrossRef](#)]
27. Kallehauge, B. Formulations and exact algorithms for the vehicle routing problem with time windows. *Comput. Oper. Res.* **2008**, *35*, 2307–2330. [[CrossRef](#)]
28. Ivancic, N.J. An Integer Programming Based Heuristic Approach to the Three Dimensional Packing Problem. Ph.D. Thesis, Case Western Reserve University, Cleveland, OH, USA, 1988.
29. Li, H.; Lim, A. A metaheuristic for the pickup and delivery problem with time windows. *Int. J. Artif. Intell. Tools* **2003**, *12*, 173–186. [[CrossRef](#)]
30. Bischoff, E.; Ratcliff, M. Issues in the development of approaches to container loading. *Omega* **1995**, *23*, 377–390. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.