Contents lists available at GrowingScience

International Journal of Industrial Engineering Computations

homepage: www.GrowingScience.com/ijiec

Using a hybrid heuristic to solve the balanced vehicle routing problem with loading constraints

Carlos A. Vega-Mejía[a], Eliana María González-Neira[b], Jairo R. Montoya-Torres[c*] and Sardar M.N. Islam[d]

[a] *Operations & Supply Chain Management Research Group, Universidad de La Sabana, Campus del Puente del Común, km 7 Autopista Norte de Bogotá D.C., Chía (Cundinamarca), Colombia*
[b] *Department of Industrial Engineering, Pontificia Universidad Javeriana, Carrera 7 40-62 Edificio José Gabriel Maldonado S.J., Bogotá D.C., Colombia*
[c] *Logistics Systems Research Group, Faculty of Engineering, Universidad de La Sabana, Campus del Puente del Común, km 7 Autopista Norte de Bogotá D.C., Chía (Cundinamarca), Colombia*
[d] *Institute for Sustainable Industries and Liveable Cities, Victoria University, P.O. Box 14425, Melbourne, Victoria, 8001, Australia*

CHRONICLE

Article history:
Received July 1 2019
Received in Revised Format
August 10 2019
Accepted August 10 2019
Available online
August 10 2019

Keywords:
Vehicle Routing Problem with Loading Constraints
Hybrid heuristic
GRASP
Clarke and Wright Savings
Practical loading and routing constraints

ABSTRACT

The Vehicle Routing Problem with Loading Constraints (VRPLC) is strongly related to real life applications in distribution logistics. It addresses the simultaneous loading and routing of vehicles, which are two crucial activities in transportation. Since treating these operations separately may result in impractical solutions, the development of applications for VRPLCs has gained the attention of researchers in recent years. Several heuristic methods have been proposed, but they consider only a limited group of practical characteristics that arise in real world situations. This study proposes a hybrid heuristic method based on the Greedy Randomized Adaptive Search Procedure (GRASP) metaheuristic and the Clarke and Wright Savings algorithm, to solve a VRPLC with several loading and routing constraints that have not been considered simultaneously before. Experimental results show that the proposed procedure produces competitive solutions in short processing times. Lastly, the impact of the added operational constraints is also analyzed.

1. Introduction

In recent years, there has been growing interest in the simultaneous determination of both the optimal routes and the packing patterns of vehicles, as this combination can assist in producing better global solutions for distribution logistics (Hokama et al., 2016). This can be carried out by modeling and solving a problem known as the Vehicle Routing Problem with Loading Constraints (VRPLC) (Zachariadis, Tarantilis, & Kiranoudis, 2013). The VRPLC is the combination of two well-known *NP-hard* problems: The Container Loading Problem (CLP) and the Vehicle Routing Problem (VRP) (Iori & Martello, 2010). Because of its potential for practical applications, the VRPLC is an emergent research stream in logistics

[*] Corresponding author
E-mail: jairo.montoya@unisabana.edu.co (J. R. Montoya-Torres)

2020 Growing Science Ltd.
doi: 10.5267/j.ijiec.2019.8.002

(Zachariadis et al., 2016), and several heuristic applications have been proposed to solve different versions of the problem.

In spite of this, there are several practical considerations, which could drive solution approaches towards more realistic scenarios, that have not been considered in the majority of solution approaches. Among the group of overlooked operational constraints, weight distribution inside the container of the vehicles and route balancing have been recognized as interesting research directions. This is because, on one hand, an improper weight distribution can increase fuel consumption (Baldi et al., 2012), and it could also impact on the safety of personnel and the safe handling of a container (Davies & Bischoff, 1999). On the other hand, achieving an efficient balance of the delivery routes (e.g. in terms of carried weight, traveled time or distance) helps to introduce aspects of fair treatment between the drivers of a transporting company (Sicilia et al., 2016).

Considering the above, the objective of this article is to present a heuristic method for solving a version of the VRPLC with characteristics not previously considered simultaneously: Container weight limit, the load-bearing strength of items, weight distribution of the load stored inside the container of the vehicle, delivery time windows, and balancing of the vehicle fleet. According to Laporte (2009), heuristic developments should be oriented towards simpler and more flexible methods, even if this means a small loss in accuracy, in order to avoid ‘over-engineered’ solution procedures. Moreover, flexibility and simplicity have also been recognized as essential attributes of good heuristics (Cordeau et al., 2002). In this regard, the proposed method is a simple streamlined procedure, with low processing computational times for both large or small instances, and the flexibility to incorporate further practical considerations. More specifically, the method is a hybrid heuristic that combines a Greedy Randomized Adaptive Search Procedure (GRASP) heuristic and a Clarke and Wright Savings (CWS) algorithm. This hybrid heuristic expands on the previous work by Vega-Mejía and Montoya-Torres (2017) by providing a more detailed explanation of the solution procedure and a deeper analysis of the computational results and implications of the considered practical constraints. It is expected that the proposed heuristic procedure serves as a starting point to represent real life situations in distribution operations more precisely.

The remainder of the article is organized as follows. Section 2 provides a brief review of commonly used heuristic approaches and previously considered loading and routing constraints. Section 3 presents a more formal definition of the VRPLC addressed in this article. Section 4 describes in detail the proposed hybrid heuristic. Section 5 describes the computational experiments that were carried out, providing the benchmark instances that were employed and the analysis of the experimental results. Finally, Section 6 presents some concluding remarks and provides interesting ideas for future research in VRPLC applications.

2. Background

Provided that the VRPLC is an *NP-Hard* problem, the decision to develop heuristic solutions is supported and favored in the literature about such problems. Some commonly used heuristic approaches are based on well-known metaheuristics, such as Tabu Search (TS) (e.g. Bortfeldt & Homberger, 2013; Gendreau et al., 2006), GRASP (e.g. Moura & Oliveira, 2009), Ant Colony Optimization (ACO) (e.g. Fuellerer et al., 2010), Simulated Annealing (SA) (e.g. Ceschia et al., 2013), and Variable Neighborhood Search (VNS) (e.g. Tricoire et al., 2011). According to Junqueira and Morabito (2015), these solution approaches can be grouped into three distinctive approaches. The first one is called “loading after routing”, which basically determines the delivery routes of the vehicles first, and then starts validating that the loading patterns are feasible. In the second approach, called “loading while routing”, as a delivery node is included in a delivery route, the heuristic procedure determines if the resulting packing pattern is feasible. The third approach is a combination of the other two. A fourth approach is proposed by Bortfeldt and Homberger (2013). The approach “pack first – route second” consists of first building a loading arrangement for each node in the delivery network, and then building the delivery routes, verifying that the loading arrangement for each route is feasible.

Both the CLP and VRP have been extensively studied in the literature, and recent reviews include the works by Bortfeldt and Wäscher (2013) who presented an updated classification framework for Packing Problems (PP) based on the use of the practical attributes of the problem; Montoya-Torres et al. (2015), who analyzed VRPs with multiple depots; and (Lin et al., 2014), who presented the evolution of VRP into Green VRP. The review by Caceres-Cruz et al. (2014) focused on the combination of VRPs with other activities related to transportation, to construct what they refer to as Rich VRPs (RVRP). According to their classification, the VRPLC is a type of RVRP. Regarding VRPLCs, the recent reviews by Iori and Martello (2010) and Junqueira and Morabito (2015) presented an account of the algorithmic approaches used to solve the problem. To the best of our knowledge, the most recent review on VRPLCs corresponds to the work by Vega-Mejía, Montoya-Torres and Islam (2019b), who analyzed how the different attributes of the problem (i.e. objective functions and operational constraints) could be realigned towards sustainable transportation applications.

Some of the previous studies argue for the necessity of including several practical characteristics when solving packing or routing problems. However, Bortfeldt and Wäscher (2013) concluded from their review work on Packing Problems (PP), that many of the practical constraints originally described by Bischoff and Ratcliff (1995) had been neglected in PP studies. Moreover, Iori and Martello (2010) and Junqueira and Morabito (2015) suggested the inclusion of several operational attributes of the VRPLC (e.g. split deliveries, weight distribution, route balancing, time windows, pickup and delivery) as future research directions in the development of solution methods. In their review, Junqueira and Morabito (2015) showed that studies have mostly concentrated on ten practical constraints: (i) Rotation of items, (ii) vertical stability, (iii) Last In – First Out (LIFO) loading/unloading, (iv) fragility of items, (v) box to pallets and pallets into vehicles, (vi) weight related constraints, (vii) time windows, (viii) time-constrained routes, (ix) pickup and delivery, (x) and split deliveries. However, the studies they analyzed considered only half of these attributes, at the most. Similar findings can be observed in the previous review works (e.g. Vega-Mejía et al., 2019b). To the best of our knowledge, the practical constraints considered in the present study have not been considered simultaneously in heuristic solution procedures for VRPLCs before.

Other recent studies seem to follow the trend described by Junqueira and Morabito (2015). For instance, Dominguez, Juan and Faulin (2014) considered as practical constraints the weight limit of the container of the vehicles, LIFO loading/unloading, and the possibility of rotating the items, in the minimization of the transportation costs of a 2-Dimensional (2D) VRPLC. To solve the problem, the authors employed a Random-Biased CWS algorithm, where the packing conditions were checked, as the routes were merged (i.e. loading while routing). This prevented the generation of any infeasible solutions. The heuristic method proposed by Zhang et al. (2015), aimed at minimizing fuel consumption in a CVRP with 3-Dimensional (3D) items, considers sufficient vertical support and the fragility of items, LIFO conditions, container weight limits and a heterogeneous vehicle fleet. The authors implemented an Evolutionary Local Search (ELS), whose initial solution was generated using a CWS algorithm for the routing part, and sorting rules of the items based on their fragility, LIFO order, vertical support and volume, for the packing problem.

Bortfeldt, Hahn, Männel and Mönch (2015) proposed two hybrid algorithms to analyze the impact of the neighborhood structure on the quality of the solution of a 3D VRPLC with the objective of minimizing the total traveled distance. In the first algorithm, the routing sub-problem is solved by an Adaptive Large Neighborhood Search (ALNS). In the second algorithm, the routing problem is solved employing a VNS, whose initial solution is generated by a CWS algorithm. In both hybrid algorithms, the packing procedure is performed with a Tree Search Algorithm (TSA). As was the case in the study by Zhang et al. (2015), the items were tagged as either fragile or non-fragile.

Dominguez et al. (2016c) proposed a multi-start Biased-Randomized CWS algorithm to minimize the total costs of a 2D VRPLC, where the vehicle fleet consists of heterogeneous vehicles. As practical considerations, the rotation of the boxes was allowed and there was a limit on the weight a vehicle could transport. The authors suggested that other practical routing aspects such as pick-up and delivery, time windows, and stochastic demands may offer interesting research directions. In related studies, Dominguez et al. (2016b) and Dominguez et al. (2016a) used biased randomization based algorithms and a CWS algorithm to solve 2D VRPLCs with the objective of minimizing the total distribution costs, using heterogeneous and homogeneous vehicle fleets, respectively. Dominguez, Juan, de la Nuez, et al. (2016) used an Iterated Local Search (ILS) to handle operational constraints such as the rotation of boxes, the weight capacity of the transporting vehicles, and LIFO loading/unloading. Dominguez, Guimarans, et al. (2016) employed an LNS to solve the problem, which considered box rotations, LIFO loading/unloading, and backhauls. In the three studies, the cargo arrangements are checked every time two routes are merged by the CWS. Continuing along this line of research, more recently Guimarans et al. (2018) minimized the total travel time in a 2D VRPLC employing a simheuristic approach (see Juan et al., 2015) that combined Monte Carlo Simulation and a biased randomized ILS. The authors considered some of the practical constraints mentioned in previous studies and added stochastic travel times to represent changing traffic conditions. Along with the study by Guimarans et al. (2016), these are, to the best of our knowledge, the only studies that have included stochastic considerations within VRPLC formulations.

Zhang et al. (2017) proposed a hybrid heuristic that combines a Bee Colony Algorithm (BCA) with a TSA, to minimize the traveled distance in a 3D VRPLC with rotation of the boxes, vertical stability, fragility of items, the weight limit of the container, LIFO loading/unloading, and delivery time windows. Different from other studies in this brief review, the proposed hybrid heuristic employs a "pack first – route second" solution approach. As future research, the authors recommended the continuous improvement of the proposed heuristic so that it can be applied in other rich VRPs. Alinaghian, Zamanlou and Sabbagh (2017) proposed an elitist non-dominated sorting local search to minimize the total traveling time and, simultaneously, balance the weight load that the vehicles carry in a time-dependent 2D VRPLC. The authors employed a piecewise linear function to represent the concept of time dependency and claim that good quality solutions can be obtained by utilizing the proposed method, although many operational constraints, considered in previous studies, were not included (e.g. LIFO loading/unloading). It is in this regard that the authors recommended an avenue for further research on this problem.

Lastly, Koch, Bortfeldt and Wäscher (2018) proposed a hybrid heuristic approach that combines an ALNS and packing heuristics, such as bottom-left-first and touching area heuristics, to solve a 3D VRPLC with time windows and pickup and delivery conditions. Practical loading constraints are considered as well (i.e. vertical stability, rotation of items, fragile and non-fragile items, and LIFO loading/unloading). The proposed hybrid checks the feasibility of the packing arrangement of a generated route, which could be classified as a "loading while routing" approach to solve the problem. The authors suggested the consideration of different backhauls conditions as interesting topics to research further.

Based on the above and to address some of the gaps identified so far in the literature, the following sections define the VRPLC considered in this study, and the detailed explanation of how a hybrid heuristic solution method can solve it.

3. Problem definition

The VRPLC considered in this paper consists of a set of clients $\mathbf{K} = \{1, \dots, m\}$ that require the delivery of different types of items, from a set of 3D rectangular boxes $\mathbf{B} = \{1, \dots, n\}$. Each item type is defined by the dimensions BL_i, BW_i and BH_i (representing length, width and height, respectively), weight BM_i and weight bearing strength BSM_i for $\forall i \in \mathbf{B}$. The delivery task is performed using a homogeneous fleet of vehicles $\mathbf{V} = \{1, \dots, p\}$, where each vehicle has a weight capacity VM and dimensions CL, CW and CH (representing length, width and height, respectively), so that $BL_i < CL, BW_i < CW$ and $BH_i < CH$. The delivery of the items required by a client ($BK_{ik}, i \in \mathbf{B}, k \in \mathbf{K}$) must be done using only one vehicle, but one vehicle can serve multiple clients. Furthermore, each vehicle starts its delivery route at the same central depot and returns to it after delivering all the assigned orders. This central depot can be represented as client 1 in set $\mathbf{K}$. In addition, each client has a defined service time ST_k and a time window between SW_k and EW_k in which they would expect the delivery of their items to take place. Also, the time required to go from one client $k \in \mathbf{K}$ to another client $l \in \mathbf{K}$ is TT_{kl}. For simplicity, TT_{kl} is also used as the distance between clients k and l.

The objectives of this VRPLC are to minimize the total distance traveled of the vehicle fleet and possible delays, to minimize the deviation of the center of gravity of the loaded vehicle from its geometrical center, and to balance the vehicle fleet so that each vehicle carries approximately the same payload. These objectives are subject to several practical loading and routing constraints, such as vertical stability, the load bearing strength of the items, the weight capacity of the transporting vehicle, the sequence for loading/unloading (i.e. LIFO), the weight distribution inside the vehicle container, delivery time windows, and determining a balanced vehicle fleet. To better illustrate this, the next section presents a Non-Linear Mixed Integer Program (NLMIP) for the problem.

3.1.NLMIP for the VRPLC

The following NLMIP model has been presented by Vega-Mejía, Montoya-Torres and Islam (2019a), who based their model on the MIP model proposed by Junqueira et al. (2013). For practical purposes, the model by Vega-Mejía et al. (2019a) is reproduced here in a summarized manner.

3.1.1. Sets

Apart from the sets mentioned previously, the following sets are used in the formulation. Set $\mathbf{S} = \{0, \dots, m\}$ represents the different transitions on the route of a vehicle. Assuming that BL_i, BW_i, BH_i have integer values $\forall i \in \mathbf{B}$, the sets $\mathbf{X} = \left\{0, \dots, VL - \min_{\forall i \in \mathbf{B}}(BL_i)\right\}, \mathbf{Y} = \left\{0, \dots, VW - \min_{\forall i \in \mathbf{B}}(BW_i)\right\}$ and $\mathbf{Z} = \left\{0, \dots, VH - \min_{\forall i \in \mathbf{B}}(BH_i)\right\}$ represent the available positions in which boxes can be placed inside the vehicles' containers. Additional sets **XNP** and **ZNP** are also introduced to reduce the number of decision variables in the model. These sets are referred to as "normal patterns" (see Christofides & Whitlock, 1977; Cui, 2007; Junqueira et al., 2013). Since the "normal patterns" limit the placement positions on each axis, a "normal pattern" is not defined for **Y**, to allow the improvement of the center of gravity.

3.1.2. Variables

Binary variables a_{xyz}^{ikvs} are defined to determine the placement of the boxes inside the vehicles, and d_{kl}^{vs} to specify the delivery route of each vehicle, with $i \in \mathbf{B}, k, l \in \mathbf{K}, s \in \mathbf{S}, v \in \mathbf{V}, x \in \mathbf{XNP} | x \leq VL - BL_i, y \in \mathbf{Y} | y \leq VW - BW_i, z \in \mathbf{ZNP} | z \leq VH - BH_i$ (see Vega-Mejía et al., 2019a). Variables b_{xyz}^{v} and h_{xyz}^{v}, with $x \in \mathbf{X} | x \leq VL - BL_i, y \in \mathbf{Y} | y \leq VW - BW_i, z \in \mathbf{Z} | z \leq VH - BH_i$, are used as variables to handle the vertical stability and LIFO constraints. Variables c_{kv}, g_{kv} and f_{kv} are used to determine the departure, arrival and tardiness, respectively, of vehicle $v \in \mathbf{V}$ when stopping at the location of client $k \in \mathbf{K}$. The variables $vload_v$ are used to calculate the weight of the load that vehicle $v \in \mathbf{V}$ carries when it leaves the central depot. The maximum and minimum weights carried by the vehicles are represented by variables $maxvload$ and $minvload$, respectively. And variables $cogy_{vs}, devcog_{vs}$ and $vloadstage_{vs}$ are used for determining the geometrical location of the center of gravity, how much it deviates from the mid-point of the width of the container, and the weight of vehicle $v \in \mathbf{V}$ in stage $s \in \mathbf{S}$, respectively. Finally, a_{xyz}^{ikvs} and d_{kl}^{vs} are binary variables, while the rest are real positive variables.

3.1.3. Model formulation

The following is the model presented by Vega-Mejía et al. (2019a). This section only presents the model formulation and a brief explanation. For a full detailed explanation of the model and computational experiments, the reader is referred to the study by Vega-Mejía et al. (2019a).

$$\min z_1 = \sum_{\forall k \in \mathbf{K}} \sum_{\forall l \in \mathbf{K}} \sum_{\forall v \in \mathbf{V}} \sum_{\forall s \in \mathbf{S}} TT_{kl} d_{kl}^{vs} \qquad (1)$$

$$\min z_2 = \sum_{\forall k\in\mathbf{K}} \sum_{\forall v\in\mathbf{V}} f_{kv} \tag{2}$$

$$\min z_3 = maxvload - minvload \tag{3}$$

$$\min z_4 = \sum_{\forall v\in\mathbf{V}} \sum_{\forall s\in\mathbf{S}} devcog_{vs} \tag{4}$$

$$\sum_{l\in\mathbf{K},l\neq k} \sum_{\forall v\in\mathbf{V}} \sum_{\forall s\in\mathbf{S}} d_{kl}^{vs} = 1 \qquad k\in\mathbf{K}\backslash\{0\} \tag{5}$$

$$\sum_{\forall l\in\mathbf{K}} \sum_{\forall v\in\mathbf{V}} \sum_{s\in\mathbf{S}\backslash\{1\}} s\cdot d_{kl}^{vs} - \sum_{\forall l\in\mathbf{K}} \sum_{\forall v\in\mathbf{V}} \sum_{\forall s\in\mathbf{S}} s\cdot d_{lk}^{vs} = 1 \qquad k\in\mathbf{K}\backslash\{0\} \tag{6}$$

$$\sum_{\forall l\in\mathbf{K}} d_{kl}^{v(s+1)} - \sum_{\forall l\in\mathbf{K}} d_{lk}^{vs} = 0 \qquad k\in\mathbf{K}\backslash\{0\}, \forall v\in\mathbf{V}, s\in\mathbf{S}\backslash\{m\} \tag{7}$$

$$\sum_{l\in\mathbf{K}\backslash\{0\}} \sum_{\forall s\in\mathbf{S}} d_{0l}^{vs} \leq 1 \qquad \forall v\in\mathbf{V} \tag{8}$$

$$\sum_{\forall v\in\mathbf{V}} \sum_{\forall s\in\mathbf{S}} \sum_{\forall x\in\mathbf{XNP}|x\leq VL-BL_i} \sum_{\forall y\in\mathbf{Y}|y\leq VW-BW_i} \sum_{\forall z\in\mathbf{ZNP}|z\leq VH-BH_i} a_{xyz}^{ikvs} = BK_{ik} \\ \forall i\in\mathbf{B}, \forall k\in\mathbf{K}, BK_{ik}>0 \tag{9}$$

$$\sum_{\forall i\in\mathbf{B}|BK_{ik}>0} \sum_{\forall x\in\mathbf{XNP}|x\leq VL-BL_i} \sum_{\forall y\in\mathbf{Y}|y\leq VW-BW_i} \sum_{\forall z\in\mathbf{ZNP}|z\leq VH-BH_i} a_{xyz}^{ikvs} = \sum_{\forall i\in\mathbf{B}|BK_{ik}>0} \sum_{l\in\mathbf{K},l\neq k} BK_{ik}\cdot \\ d_{lk}^{vs} \qquad k\in\mathbf{K}\backslash\{0\}, \forall v\in\mathbf{V}, s\in\mathbf{S}\backslash\{m\} \tag{10}$$

$$\sum_{\forall i\in\mathbf{B}} \sum_{\forall k\in\mathbf{K}\backslash\{0\}} \sum_{l\in\mathbf{K},l\neq k} \sum_{\forall s\in\mathbf{S}\backslash\{0\}} BK_{ik}\cdot BL_i\cdot BW_i\cdot BH_i\cdot d_{kl}^{vs} \leq VL\cdot VW\cdot VH \qquad \forall v\in\mathbf{V} \tag{11}$$

$$\sum_{(i,k)\in\mathbf{B}\times\mathbf{K}|BK_{ik}>0} \sum_{s\in\mathbf{S}\backslash\{m\}} \sum_{\forall x\in\mathbf{XNP}|x'-BL_i+1\leq x\leq\min(x',VL-BL_i)} \sum_{\forall y\in\mathbf{Y}|y'-BW_i+1\leq y\leq\min(y',VW-BW_i)} \\ \sum_{\forall z\in\mathbf{ZNP}|z'-BH_i+1\leq z\leq\min(z',VH-BH_i)} a_{xyz}^{ikvs} \leq 1 \qquad \forall v\in\mathbf{V}, \forall x'\in\mathbf{XNP}, \forall y'\in\mathbf{Y}, \forall z'\in\mathbf{ZNP} \tag{12}$$

$$\sum_{(i,k)\in\mathbf{B}\times\mathbf{K}|BK_{ik}>0} \sum_{s\in\mathbf{S}\backslash\{m\}} \sum_{\forall x\in\mathbf{XNP}|x'-BL_i+1\leq x\leq\min(x',VL-BL_i)} \sum_{\forall y\in\mathbf{Y}|y'-BW_i+1\leq y\leq\min(y',VW-BW_i)} \\ \sum_{\forall z\in\mathbf{ZNP}|z'-BH_i+1\leq z\leq\min(z',VH-BH_i)} a_{xyz}^{ikvs} = b_{x'y'z'}^{v} \qquad \forall v\in\mathbf{V}, \forall x'\in\mathbf{X}, \forall y'\in\mathbf{Y}, \forall z'\in\mathbf{Z} \tag{13}$$

$$z\cdot b_{xyz}^{v} \leq \sum_{z'=0}^{z-1} b_{xyz'}^{v} \qquad \forall v\in\mathbf{V}, \forall x\in\mathbf{X}, \forall y\in\mathbf{Y}, \forall z\in\mathbf{Z}, z\geq 1 \tag{14}$$

$$\sum_{(i,k)\in\mathbf{B}\times\mathbf{K}|BK_{ik}>0} \sum_{s\in\mathbf{S}\backslash\{m\}} \sum_{\forall x\in\mathbf{XNP}|x'-BL_i+1\leq x\leq\min(x',VL-BL_i)} \sum_{\forall y\in\mathbf{Y}|y'-BW_i+1\leq y\leq\min(y',VW-BW_i)} \\ \sum_{\forall z\in\mathbf{ZNP}|z'-BH_i+1\leq z\leq\min(z',VH-BH_i)} s\cdot a_{xyz}^{ikvs} = h_{x'y'z'}^{v} \qquad \forall v\in\mathbf{V}, \forall x'\in\mathbf{X}, \forall y'\in\mathbf{Y}, \forall z'\in\mathbf{Z} \tag{15}$$

$$h_{xyz}^{v} \geq h_{x'yz'}^{v} \qquad \forall v\in\mathbf{V}, x\in\left\{0,\dots,VL-\min_{\forall i\in\mathbf{B}}(BL_i)-1\right\}, \forall y\in\mathbf{Y}, \\ z\in\left\{0,\dots,VH-\min_{\forall i\in\mathbf{B}}(BH_i)-1\right\}, x'\in\left\{x+1,\dots,VL-\min_{\forall i\in\mathbf{B}}(BL_i)\right\}, z'\in\left\{z,\dots,VH-\min_{\forall i\in\mathbf{B}}(BH_i)\right\} \tag{16}$$

$$h_{xyz}^{v} \geq h_{xyz'}^{v} \qquad \forall v\in\mathbf{V}, \forall x\in\mathbf{X}, \forall y\in\mathbf{Y}, z\in\left\{0,\dots,VH-\min_{\forall i\in\mathbf{B}}(BH_i)-1\right\}, \\ z'\in\left\{z+1,\dots,VH-\min_{\forall i\in\mathbf{B}}(BH_i)\right\} \tag{17}$$

$$\sum_{(j,k)\in\mathbf{B}\times\mathbf{K}|BK_{jk}>0} \sum_{u\in\mathbf{S}\backslash\{m\}} \sum_{x''\in\mathbf{XNP}|x'-BL_j+1\leq x''\leq\min(x',VL-BL_j)} \sum_{y''\in\mathbf{Y}|y'-BW_j+1\leq y''\leq\min(y',VW-BW_j)} \\ \sum_{z''\in\mathbf{ZNP}|z''\leq VH-\quad j} \left(\frac{BM_j}{BL_j\cdot BW_j}\right) a_{x''y''z''}^{jkvu} \leq \sum_{(i,k)\in\mathbf{B}\times\mathbf{K}|BK_{ik}>0} \sum_{s\in\mathbf{S}\backslash\{m\}} \sum_{x\in\mathbf{XNP}|x'-BL_i+1\leq x\leq\min(x',VL-BL_i)} \\ \sum_{y\in\mathbf{Y}|y'-BW_i+1\leq y\leq\min(y',VW-BW_i)} \sum_{z\in\mathbf{ZNP}|z'-BH_i+1\leq z\leq\min(z',VH-BH_i)} \left(\frac{BSM_i}{BL_i\cdot BW_i}\right) a_{xyz}^{ikvs} \\ \forall v\in\mathbf{V}, \forall x'\in\mathbf{XNP}, \forall y'\in\mathbf{Y}, \forall z'\in\mathbf{ZNP} \tag{18}$$

$$c_{kv} \geq SW_k\cdot\left(\sum_{\forall l\in\mathbf{K},l\neq k} \sum_{\forall s\in\mathbf{S}} d_{lk}^{vs}\right) \qquad k\in\mathbf{K}\backslash\{0\}, \forall v\in\mathbf{V} \tag{19}$$

$$c_{kv} - f_{kv} \leq EW_k\cdot\left(\sum_{\forall l\in\mathbf{K},l\neq k} \sum_{\forall s\in\mathbf{S}} d_{lk}^{vs}\right) \qquad k\in\mathbf{K}\backslash\{0\}, \forall v\in\mathbf{V} \tag{20}$$

$$g_{0v} = \sum_{k\in\mathbf{K}\backslash\{0\}} (SW_k - TT_{0k})\cdot d_{0k}^{v1} \qquad \forall v\in\mathbf{V} \tag{21}$$

$$g_{kv} \geq c_{kv} + f_{kv} + ST_k\cdot\left(\sum_{\forall l\in\mathbf{K},l\neq k} \sum_{\forall s\in\mathbf{S}} d_{lk}^{vs}\right) \qquad k\in\mathbf{K}\backslash\{0\}, \forall v\in\mathbf{V} \tag{22}$$

$$c_{lv} \geq g_{kv} + TT_{kl} - M\cdot\left(1 - \sum_{\forall s\in\mathbf{S}} d_{kl}^{vs}\right) \qquad (k,l)\in\mathbf{K}, k\neq l, \forall v\in\mathbf{V} \tag{23}$$

$$vload_v = \sum_{(i,k)\in\mathbf{B}\times\mathbf{K}|BK_{ik}>0} \sum_{s\in\mathbf{S}\backslash\{m\}} \sum_{\forall x\in\mathbf{XNP}|x\leq VL-BL_i} \sum_{\forall y\in\mathbf{Y}|y\leq VW-BW_i} \sum_{\forall z\in\mathbf{ZNP}|z\leq VH-BH_i} a_{xyz}^{ikvs} \qquad \forall v\in\mathbf{V} \tag{24}$$

$$vload_v \leq VM \qquad \forall v\in\mathbf{V} \tag{25}$$

$$vload_v \leq maxvload \qquad \forall v\in\mathbf{V} \tag{26}$$

$$vload_v \geq minvload \qquad \forall v\in\mathbf{V} \tag{27}$$

$$\sum_{(i,k)\in\mathbf{B}\times\mathbf{K}|BK_{ik}>0} \sum_{s'=s}^{m-1} \sum_{\forall x\in\mathbf{XNP}|x\leq VL-BL_i} \sum_{\forall y\in\mathbf{Y}|y\leq VW-BW_i} \sum_{\forall z\in\mathbf{ZNP}|z\leq VH-BH_i} BM_i\cdot a_{xyz}^{ikvs} = \\ vloadstage_{vs} \qquad \forall v\in\mathbf{V}, s\in\mathbf{S}\backslash\{m\} \tag{28}$$

$$cogy_{vs} \cdot vloadstage_{vs} = \sum_{(i,k)\in \mathbf{B}\times\mathbf{K}|BK_{ik}>0} \sum_{s'=s}^{m-1} \sum_{\forall x\in \mathbf{XNP}|x\leq VL-BL_i} \sum_{\forall y\in \mathbf{Y}|y\leq VW-BW_i} \sum_{\forall z\in \mathbf{ZNP}|z\leq VH-BH_i} BM_i \cdot \left(y+\frac{BW_i}{2}\right)\cdot a_{xyz}^{ikvs'} \quad \forall v\in \mathbf{V}, s\in \mathbf{S}\backslash\{m\} \tag{29}$$

$$\frac{VW}{2} - cogy_{vs} \leq devcog_{vs} \quad \forall v\in \mathbf{V}, s\in \mathbf{S}\backslash\{m\} \tag{30}$$

$$cogy_{vs} - \frac{VW}{2} \leq devcog_{vs} \quad \forall v\in \mathbf{V}, s\in \mathbf{S}\backslash\{m\} \tag{31}$$

Particularly, (1) is defined to minimize the total distance traveled, (2) minimizes the total tardiness, in (3) the vehicle fleet is balanced by minimizing the difference between the most and the least loaded vehicles, and (4) minimizes the total deviation of the Center Of Gravity (COG) of the loaded vehicles from the mid-point of the width of the container. These objectives are subject to the following constraints: (5) Establish that the items of a client have to be delivered by only one vehicle, (6) and (7) guarantee network flow feasibility, (8) forces vehicles to start their travel from the central depot, (9) cover the demand of the clients, (10) guarantee that the items of a client will be transported by a single vehicle, (11) and (12) refer to the basic packing constraints explained by Wäscher et al. (2007), (13) and (14) enforce full vertical support for boxes not placed on the floor of the container of the vehicle, (15)-(17) are used for the LIFO loading/unloading of items, (18) controls the weight bearing strength of items, (19) and (20) consider the delivery time windows. Constraints (21)-(23) determine each vehicle's departure time from, and arrival time at, the location of a client. Finally, constraints (24) and (25) control the maximum weight that can be loaded in each vehicle, (26) and (27) determine the most and least loaded vehicles, and (28)-(31) are used to examine weight distribution inside the container of each vehicle.

4. The hybrid heuristic

As previously mentioned, the hybrid heuristic expands on the one proposed in the work by Vega-Mejía and Montoya-Torres (2017). The hybrid heuristic presented in this section is based on a "pack first – route second" approach was selected. The rationale behind this decision was the combination of the practical loading and routing constraints of the problem, and the techniques used in previous studies to address them. For instance, Eley (2002) dealt with weight distribution by grouping items in order to build blocks and then swapping these blocks with others to obtain a better COG of the loaded container. García-Cáceres, Vega-Mejía and Caballero-Villalobos (2011) divided the loaded container into walls, which were swapped with one another and then reflected relative to their mid-point to minimize the distance of the COG to the geometrical center of the container. By constructing a packing arrangement for the items of each client prior to the construction of any vehicle routes, the process of rearranging the blocks, that do not interlock with others, to improve the COG of the container of a vehicle is simplified. This approach is based on the one presented by Lim, Ma, Qiu and Zhu (2013), in which the blocks are prevented from interlocking in order to facilitate the process of improving the weight distribution inside the packed container.

Another reason for using the "pack first – route second" approach has to do with the considerations of some of the loading constraints. Since split deliveries are not allowed (i.e. items of a client must be delivered by a single vehicle), the reliability and duration of the distribution process could be improved if items of the same client are placed close to each other inside the container of the vehicle. Building a cargo pattern for each client, that groups all their items into a single block before the delivery route is planned, guarantees this. This block arrangement could also guarantee that an item being unloaded in stage s of a route, would not be blocked by another item that has to be delivered at a later stage $s'(s' > s)$. Hence, the total time taken to accomplish all the deliveries could be reduced as rearrangement of items is prevented after each stop. Furthermore, balancing the carried load of the vehicles involves moving items from one vehicle to another. Since there are no split deliveries, the complete set of items for that client should be moved from one vehicle to another. A predefined packing pattern for each client would greatly simplify this analysis and would avoid a complete reconstruction of the loading arrangement of a vehicle.

To consider what has been stated until this point, the proposed hybrid heuristic consists of three stages (see Fig. 1). Stage 1 generates the blocks for each client. Considering that the blocks impact the number of required vehicles, these should be formed to use the space inside the container efficiently. In this sense, the building of a block is reduced to solving a 3D Strip Packing Problem (3D SPP), in which the objective is to minimize the surface area in which all the items are packed together (i.e. strip). For this task a GRASP metaheuristic is employed. Stage 2 defines the routing for each vehicle and packs the generated blocks into the vehicles. For the routing task a CWS algorithm is used to solve a VRP with Time Windows (VRPTW). For the packing of the blocks into the vehicles and to facilitate the exchange of blocks between vehicles, no weight will be placed on top of the blocks. This reduces the packing of the vehicle to a 2D PP. This problem is solved using the GRASP metaheuristic from the first stage. Stage 3 consists of balancing the vehicle fleet by employing a simple local search procedure that swaps blocks between vehicles, while at the same time reducing the factors of traveled distance and total tardiness. Finally, the distribution of the weight inside the container of each vehicle is also addressed.

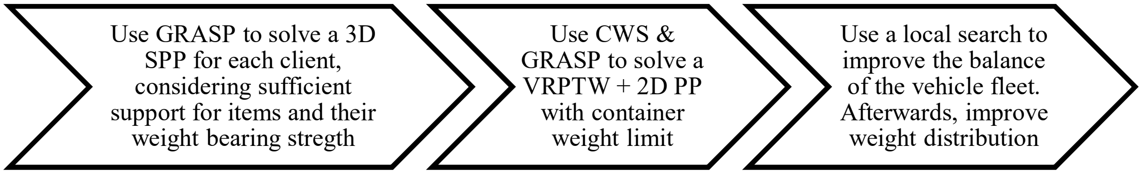

Use GRASP to solve a 3D SPP for each client, considering sufficient support for items and their weight bearing stregth
Use CWS & GRASP to solve a VRPTW + 2D PP with container weight limit
Use a local search to improve the balance of the vehicle fleet. Afterwards, improve weight distribution

Fig. 1. Basic process of the hybrid heuristic – Adapted from Vega-Mejía and Montoya-Torres (2017)

The following sections explain in more detail each of the procedural stages of the hybrid heuristic.

4.1. Stage 1: A GRASP approach to solve a 3D SPP

GRASP is an iterative process consisting of two phases: constructive and local search (Resende & Ribeiro, 2010). The following paragraphs explain how the construction phase and the local search phase of GRASP are applied to build the loading arrangements for each client by solving a 3D SPP, while considering sufficient support for those items not placed on the floor of the container and the weight bearing strength of the items.

4.1.1. Constructive phase

The construction phase oversees the generation of a feasible solution for the problem. Previous solution approaches for PPs are based on sorting the items according to some of their attributes, for instance their area, volume or weight (e.g. Egeblad et al., 2010; Eley, 2002), and then using a placing strategy (e.g. best fit, left bottom fit, first fit) to assign an item to a position or corner inside the transporting container. However, sorting the items according to such basic attributes may result in an improper load when the weight distribution inside the container is considered (Lim et al., 2013). With this in mind, Lim et al. (2013) defined a constructive phase for GRASP that identifies the available free spaces in a container after an item is stored. The construction phase for the proposed GRASP is based on this notion and on the identification of insertion points described in the work by Zachariadis et al. (2013).

There are two vital components in this phase of GRASP. A utility function which evaluates each of the elements that may become part of the feasible solution, and a Restricted Candidates List (RCL) which stores those elements with a utility function whose value lies in the interval $[L, L + (U - L)\alpha]$ (García-Cáceres et al., 2011), where L and U are the lower and upper values of the utility function for all the elements, and α is a random number between 0 and 1.

The utility function for solving the 3D SPP for the first stage of the hybrid heuristic, which was first presented in the work by Vega-Mejía and Montoya-Torres (2017), is as follows:

$$c(i) = \frac{average\ number\ of\ valid\ insertion\ points\ after\ placing\ item\ i}{weight\ bearing\ strength\ of\ item\ i}$$

Where $c(i)$ is the utility function associated with item i. To better understand this expression, consider the situation depicted in Fig. 2(a). A box of type A (A_1) has already been placed in the bottom left corner of the container. Notice that the figure also shows the possible insertion points generated by the box already stored inside the container. The constructive phase will have to select the next item to place inside the container, and for this there are two possible items to select: One of type A (A_2) and one of type B.

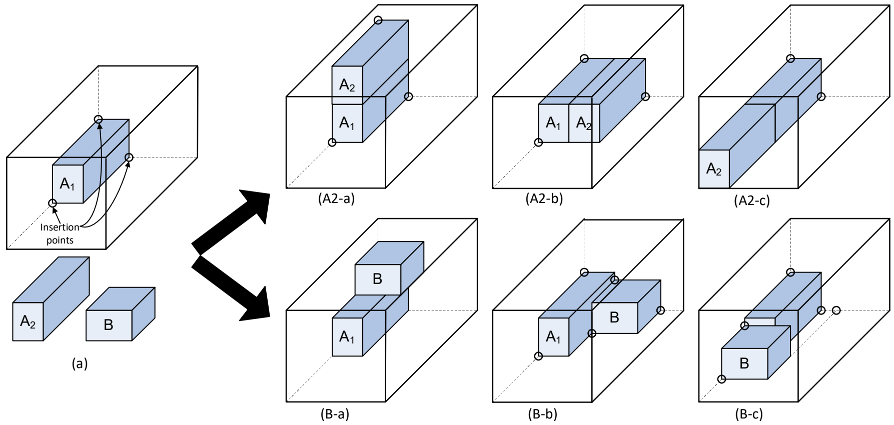


Fig. 2. Packing situation

The average number of valid insertion points for the two items is determined by placing each item on each available insertion point, and then counting the number of feasible insertion points that are generated after placing the item. Positioning the item is done by placing its bottom left corner on the insertion point. Finally, the average is computed by dividing the total number of insertion points by the number of stable loading arrangements, without yet considering the weight bearing constraints. In Fig. 2, elements (A2-a,b,c) and (B-a,b,c) show the insertion points generated by placing items A_2 and B, respectively. Notice that in element (B-a) no insertion points were generated. This is because item B would not have sufficient support if it were placed on top of item A_1.

Assuming that the items of type A have a greater weight bearing strength than those of type B ($BSM_A > BSM_B$), the respective values of the utility function for each item would be as follows:

$$c(A_2) = \frac{\left(\frac{3+3+2}{3}\right)}{BSM_{A_2}} = \frac{8}{3BSM_{A_2}} \qquad c(B) = \frac{\left(\frac{5+5}{2}\right)}{BSM_B} = \frac{5}{BSM_B}$$

It would follow then that $c(A_2) < c(B)$, and hence the RCL would be populated with the items whose utility function is within the interval $\left[\frac{8}{3BSM_{A_2}}, \frac{8}{3BSM_{A_2}} + \left(\frac{5}{BSM_B} - \frac{8}{3BSM_{A_2}}\right)\alpha\right]$. If $\alpha < 1$, item A_2 would be included in the RCL and would be selected to become part of the solution. This would ultimately mean that the utility function would have guided the selection of the item that produces a more homogenous cargo pattern. This is in the same vein as the idea proposed by Eley (2002), that items of the same type should be placed together to build cargo patterns with a reduced number of empty spaces. Moreover, the utility function also aids to populate the RCL with the items that offer more weight bearing resistance, so that other items can be placed on top of them.

Once an item is randomly selected from the RCL, it is assigned to a proper insertion point or corner. For instance, Zhang et al. (2015) sorted the available corners or spaces according to their coordinates (Z, Y, X), while Gendreau et al. (2006) preferred a (X, Z, Y) sorting. The first approach guides the filling of the container from the ground up. However, due to the objective of packing as many items in a reduced space while utilizing the whole of the container in the best way possible, a (X, Z, Y) sorting might be more appropriate. In essence, this approach is similar to a wall building approach (see Pisinger, 2002). This selection process is repeated until all the items have been placed inside the container. However, the weight-bearing and sufficient vertical support constraints introduce additional complexities to the problem, and unfeasible solutions are possible. In this sense, if at a particular moment in the constructive phase an item cannot be placed, a rotation of the item in the plane $X - Y$ is allowed. If this does not work, the constructive phase can relax the vertical support and weight bearing constraints. After all items have been placed, the constructive phase ends by determining the surface area of the generated block as shown in Fig. 3.

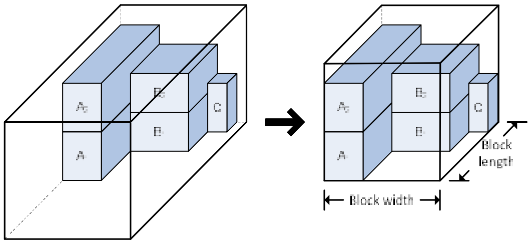


Fig. 3. Generated block

Finally, Fig. 4 shows the pseudo-code for this phase, which is based on that presented by García-Cáceres et al. (2011).

```
PROCEDURE Constructive Phase
PARAMETERS
    $\alpha$: Numeric value between 0 and 1
    $E$: Client items
    $c(\cdot)$: Utility function
VARIABLES
    $x$: Initial solution
    $RCL$: Restricted candidates list
BEGIN PROCEDURE
    $x \leftarrow \emptyset$
    Evaluate utility function $c(e), \forall e \in E$
    WHILE $E \neq \emptyset$
        $c_* \leftarrow \min\{c(e) | e \in E\}$
        $c^* \leftarrow \max\{c(e) | e \in E\}$
        $RCL \leftarrow \{e \in E \mid c(e) \leq c_* + \alpha(c^* - c_*)\}$
        Choose an element $s$ at random from the $RCL$
        IF $x \cup \{s\}$ is not feasible THEN
            Rotate $s$
            IF $x \cup \{s\}$ is not feasible THEN
                Relax weight bearing and sufficient support
                IF $x \cup \{s\}$ is not feasible THEN
                    RETURN unable to find feasible solution
                END IF
            END IF
        END IF
        $x \leftarrow x \cup \{s\}$
        Remove element $s$ from $E$
        Evaluate utility function $c(e), \forall e \in E$
    END WHILE
    Determine block size of $x$
    RETURN $x$
END PROCEDURE
```

Fig. 4. GRASP Constructive Phase for 3D SPP

4.1.2. Local search phase

The second phase of the GRASP method consists of a local search that tries to improve the size of the block generated by the constructive phase. The improvement is done by a series of simple relocation moves of the items within the block, with the objective of reducing the total surface area, thus benefiting the later stages of the hybrid heuristic by providing the possibility of storing more blocks inside the container of the vehicle. Basically, the procedure finds the item placed furthest from the geometrical origin of the container, with no other items placed on top of it. The item is then relocated to other available insertion points, while checking the stability and weight bearing constraints. Fig. 5 presents this situation where item C is considered for relocation.

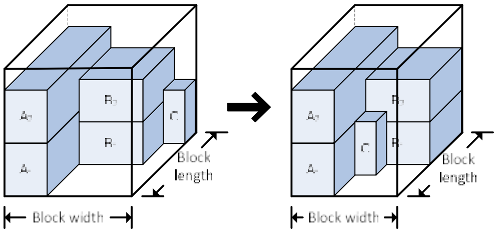


Fig. 5. Relocation of item

The item is relocated once a suitable insertion point is found, and the surface area of the block is recalculated, as shown in Fig. 5. If there is an improvement, the cargo pattern is updated by the new block, and the relocated item is blocked from being selected again for relocation. The process is repeated until no more relocation moves can be performed. When no more items can be relocated, the local search procedure calculates the center of mass of the block. This measure will ultimately be used in the final stage of the hybrid heuristic to determine the weight distribution of the loaded vehicle container. Because it is assumed that the center of mass of each item corresponds to its geometrical center, the calculation is reduced to $COGY = \left(\sum_{\forall i} M_i \left(Y_i + \frac{BW_i}{2}\right)\right) / \sum_{\forall i} M_i$, where BW_i represents the width of item i, as stated previously, and M_i and Y_i represent the weight and placement on the Y-axis of item i, respectively. Fig. 6 shows the pseudo-code for this phase.

```
PROCEDURE Local Search Phase
PARAMETERS
    $x^0$: Current solution
    $IP$: Insertion points
    $f(\cdot)$: Calculates surface area of a solution
VARIABLES
    $x$: Temporal solution
    $COGY$: Center of mass of the block along $Y$-axis
    $TL$: Tabu list that stores items that have been moved
BEGIN PROCEDURE
    $TL \leftarrow \emptyset$
    IF $x^0$ is feasible AND not relaxed THEN
        DO
            Find the item $i$ furthest from (0,0,0) and supporting no weight
            FOR EACH $p$IN $IP$
                IF $i$ can be relocated to $p$ in $x^0$ THEN
                    $x \leftarrow x^0$
                    Relocate $i$ on $p$ within $x$
                    Update $IP$
                    $TL \leftarrow TL \cup \{i\}$
                    IF $f(x) < f(x^0)$ THEN
                        $x^0 \leftarrow x$
                    END IF
                END IF
            NEXT $p$
        WHILE $i \neq \emptyset$ AND $i \notin TL$
    END IF
    $COGY$ = Calculate center of mass of $x^0$
    RETURN $x^0, COGY$
END PROCEDURE
```

Fig. 6. GRASP Local Search Phase for 3D SPP

4.2.Stage 2: A Clarke and Wright based approach to solve a two-dimensional VRP with loading constraints

The CWS algorithm has been used in previous studies to solve VRPs and VRPLCs (e.g. Tricoire et al., 2011). It is a simple procedure that creates independent routes for every node in the network, and then tries to merge the routes to minimize the required number of routes. Since each route could be assigned to one vehicle, it is implicit that the reduction of the number of routes reduces the number of required vehicles as well. In this stage, a CWS algorithm is employed to solve a 2D VRPLC, using the blocks resulting from Stage 1.

4.2.1.Route merging

The process of merging two routes in the CWS algorithm is aimed at forming a single route with a better objective value. Considering a symmetric cost (or distance or time) matrix, in a basic VRP the savings from this fusion are calculated as $S_{ij} = 2c_{i0} + 2c_{j0} - \left(c_{0j} + c_{ji} + c_{i0}\right) = c_{0j} + c_{i0} - c_{ji}$ (Mehrjerdi, 2014). Although this is useful when considering the minimization of costs (or distance or time) alone, the merging has to be modified to consider additional routing characteristics, such as time windows. In the case of this study, computation of the savings is also impacted by the available loading space and maximum weight capacity of the container of the vehicle.

For the proposed VRPLC, the merging of routes will depend on the possibility of producing a feasible loading pattern. If the blocks of the nodes belonging to the routes that are being merged cannot be accommodated inside the container of the vehicle, there is no reason for calculating other metrics such as compliance with time windows, or distance traveled, among others. Improvements for these metrics will be addressed in the last stage of the proposed heuristic, when the balancing of the vehicle fleet is performed. Nevertheless, the time windows conditions are not completely disregarded in this stage. These constraints are used to determine the order in which the blocks will be loaded into the container of the vehicle, in order to satisfy the LIFO constraints. Following the NLMIP model from Section 3, the time windows constraints are softened. This results in the consideration of tardiness, but can also ease the generation of valid cargo arrangements (Kramer et al., 2015).

To explain how two routes are merged, consider Fig. 7. Here, the container of the vehicle has an available space like the one shown in Fig. 7(a), the items of nodes i and j have been grouped into the blocks shown in Fig. 7(b) and Fig. 7(c), respectively. If the opening time window of node i is greater than that of node j, then the items of client i should be packed first, as node i could be visited later along the route. However, this implies that loading the block corresponding to node j would not be possible, unless this block were rotated. Because the rotation of the block of node j produces a feasible cargo pattern (see Fig. 7(d)) the two routes can be merged.

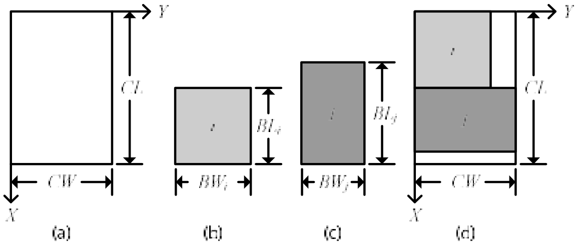


Fig. 7. Merge guaranteeing feasible loading

The packing of the blocks into the vehicle is performed by the GRASP procedure used in Stage 1. To guarantee that the constructive phase always packs the block whose associated client (node) has the highest time window first, the utility function is defined simply as the opening of the time window and $\alpha = 0$. Setting this value of α turns the GRASP procedure into a purely greedy heuristic that will select the best element (i.e. the block with the highest opening time window) at every step of the constructive phase. The result is a cargo pattern that avoids repeated loading/unloading operations since every item for a client is contained within an individual block and no block will become an obstacle when another one needs to be unloaded. Furthermore, loading the block of the client with the highest opening time window first, is aimed at reducing late deliveries. Fig. 8 shows the pseudo-code for merging the routes.

```
PROCEDURE Merge
PARAMETERS
        $R$: Set of routes
        $i, j$: Routes to merge
VARIABLES
        $x$: Cargo pattern
        $r$: Merged route
        $b$: Blocks from route
        $I$: Number of iterations for GRASP
        $\alpha$: Alpha value for GRASP
        $c(\cdot)$: Utility function for GRASP
        $f(\cdot)$: Objective function for GRASP
BEGIN PROCEDURE
        $r \leftarrow GetNodes(i) \cup GetNodes(j)$
        $b \leftarrow GetBlocks(i) \cup GetBlocks(j)$
        $I = 1 \wedge \alpha = 0$
        $c(\cdot) =$ opening time window of node
        $f(\cdot) =$ minimize wasted space inside the container of the vehicle
        $x \leftarrow GRASP(I, \alpha, b, c, f)$
        IF $x$ is feasible THEN
                Determine distance, earliness and tardiness of $r$
                Set the cargo pattern of $r$ as $x$
                Remove routes $i, j$ from $R$
                $R \leftarrow R \cup r$
        END IF
        RETURN $R$
END PROCEDURE
```

Fig. 8. Merge routes in CWS for solving a 2D VRPTWLC

4.3. Improving the solution and addressing additional operational attributes

After completing Stages 1 and 2, a partial solution is available, which integrates most of the routing and packing decisions of the discussed 3D VRPLC. The objectives of Stage 3 are to incorporate the remaining operational characteristics considered for the 3D VRPLC. The following subsections explain the processes to, first, balance the transport fleet by seeking to ensure that all the vehicles carry approximately the same payload and, second, to improve the weight distribution inside the vehicles.

4.3.1. Balancing the vehicle fleet

The first part of the Stage 3 is a simple procedure that takes the heaviest and lightest loaded vehicles and swaps blocks between them in an attempt to reduce the difference in their payloads, as represented by z_3 in the NLMIP model. For this case, the swapping moves follow a first-improve or first descent strategy. Naturally, whenever a swap is performed, the GRASP procedure must guarantee that the interchange will result in a feasible cargo pattern for both vehicles, otherwise the move is discarded. After all the block swaps between the vehicles have been examined, the procedure checks if the most and least loaded vehicles are the same. If they are not, the new heaviest and lightest loaded vehicles are selected, and the process is repeated until no more swaps are possible. Apart from addressing the balance of the vehicle fleet, the moves are aimed at minimizing the distance traveled, and the total tardiness of the system. If a swap does not improve these objectives, the move is discarded as well. The pseudo-code of the balancing procedure is shown in Fig. 9.

```
PROCEDURE Balance Vehicle Fleet
PARAMETERS
    $R$: Set of routes
    $CW$: Width of the container
VARIABLES
    $rmax, rmin$: Vehicle with most and least loaded weights
    $Nmax, Nmin$: Set for blocks of vehicles with most and least loaded weights
    $xmax, xmin$: Cargo pattern for vehicles with most and least loaded weights
    $\Delta$: Weight difference between vehicles with most and least loaded weights
    $I$: Number of iterations for GRASP
    $\alpha$: Alpha value for GRASP
    $c(\cdot)$: Utility function for GRASP
    $f(\cdot)$: Objective function for GRASP
BEGIN PROCEDURE
    $I = \alpha = 1$
    $c(\cdot)$ = opening time window of node
    $f(\cdot)$ = minimize wasted space inside the container of the vehicle
    DO
        $rmax \leftarrow GetMostLoaded(R)$
        $rmin \leftarrow GetLeastLoaded(R)$
        $Nmax \leftarrow GetBlocks(rmax)$
        $Nmin \leftarrow GetBlocks(rmin)$
        $\Delta = Weight(Nmax) - Weight(Nmin)$
        FOR EACH $(i, j) \in Nmax \times Nmin$
            $Nmax \leftarrow Nmax \cup \{j\}$
            $Nmin \leftarrow Nmin \cup \{i\}$
            Remove $i$ from $Nmax$ and $j$ from $Nmin$
            IF $Weight(Nmax) - Weight(Nmin) < \Delta$ THEN
                $xmax \leftarrow GRASP(I, \alpha, Nmax, c, f)$
                $xmin \leftarrow GRASP(I, \alpha, Nmin, c, f)$
                IF $(xmax, xmin)$ are feasible THEN
                    IF Distance, earliness and tardiness are reduced THEN
                        Update $rmax$ with $Nmax$
                        Update $rmin$ with $Nmin$
                        Improve COG $(R, CW)$
                        EXIT FOR EACH
                    END IF
                END IF
            END IF
        NEXT $(i, j)$
    WHILE $rmax \neq GetMostLoaded(R)$OR $rmin \neq GetLeastLoaded(R)$
    RETURN $R$
END PROCEDURE
```

Fig. 9. Balance the transported weight of the vehicle fleet

4.3.2. Weight distribution inside the container

Studies that have addressed this constraint have used techniques such as repositioning the blocks inside the container or swapping walls of items to reduce the distance between the COG of the loaded container and its geometrical center (see Eley, 2002; García-Cáceres et al., 2011). However, these approaches did not consider a scenario with multiple stops, which is present in VRPLCs, and brings forth a higher level of complexity, as the COG is not just calculated once, but at every stop the vehicle makes. Clearly, it would be impractical to adjust the COG of the vehicle after stopping at each delivery location, as this could result in the re-accommodation of items inside the container.

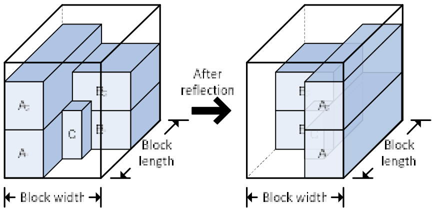


Fig. 10. Reflection of a block as presented by Vega-Mejía and Montoya-Torres (2017)

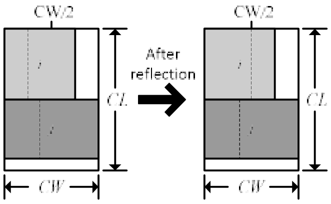


Fig. 11. Reflection of blocks inside a vehicle

Taking this into account, it is logical to argue that each vehicle must leave the central depot loaded in such a way that its COG would not vary greatly after each stop. This means that each block would have to be loaded with its center of mass as close as possible to the geometrical center of the container. In order to do this, the weight distribution proposed here considers the idea presented by García-Cáceres et al. (2011) of reflecting a wall (a block in this particular case) upon its middle point. The basic concept is shown in Fig. 10 considering the example from Fig. 5. This transformation could then be extended to all blocks assigned to a vehicle. Using Fig. 7(d), a possible improvement of the COG of a vehicle would be similar to what is shown in Fig. 11, where the center of mass of each block is represented by the dotted lines. It is expected that, before reflecting block i, the COG of the vehicle will stay to the left of the midpoint of the width of the container after each stop along the route. After reflecting block i, its center of mass would match the midpoint of the container's width after delivering the items forming block j, hence producing a better weight distribution inside the vehicle throughout the delivery route. The process for improving the COG is applied to each of the generated routes, after all the other operational considerations have been resolved (see Fig. 12). If there is an improvement, then the y-positions of every item in the reflected block are reflected as well.

```
PROCEDURE Improve COG
PARAMETERS
        $R$: Set of routes
        $CW$: Width of the container
VARIABLES
        $N$: Set for blocks
        $B$: Set for items of blocks
BEGIN PROCEDURE
        FOR EACH $r$ IN $R$
                $N \leftarrow GetBlocks(r)$
                FOR EACH $n$ IN $N$
                        IF $\left|\frac{CW}{2} - (CoordY(n) + COGY_n)\right| > \left|\frac{CW}{2} - (CoordY(n) + Width(n) - COGY_n)\right|$ THEN
                                $COGY_n = Width(n) - COGY_n$
                                $B \leftarrow GetItems(n)$
                                FOR EACH $b$ IN $B$
                                        $CoordY(b) = CW - CoordY(b) - Width(b)$
                                NEXT $b$
                        END IF
                NEXT $n$
        NEXT $r$
        RETURN $R$
END PROCEDURE
```

Fig. 12. Improvement of weight distribution

5. Computational experiments

The proposed hybrid heuristic was coded in C# (C-Sharp) and tested on a laptop running Windows 8 operating system, using an Intel Core-i7 processor and 8GB of RAM. For all tests, the value of the α parameter for GRASP in the first phase of the proposed hybrid, was originally set at 0.1 for each client, and increased by 0.1 every time the procedure was unable to construct a block for the items of a client. If parameter α reaches the value of $\alpha = 1$ without being able to find a feasible loading plan, the hybrid heuristic would halt the process and indicate that it was unable to generate a feasible solution for the problem. The following sections describe the problem instances used to perform the computational experiments and present the solution values generated by the heuristic algorithm.

5.1. Used problem instances

The heuristic was tested with the four classes of problems described in the work by Bortfeldt and Homberger (2013), namely GI_I 1, GI_I2, GII_I1 and GII_I2, grouping a total of 46 problem instances each employing containers of size $(CL: 1360, CW: 245, CH: 300)$. These instances were originally used in the study by Moura and Oliveira (2009) and are available under the name "CLP and VRPTW integration", at web.fe.up.pt/~esicup/datasets. However, some of the parameters included in the proposed VRPLC are not present in the instances (i.e. the weight and the weight bearing strength of each item, and the weight capacity of the container). To account for this, the weight of each item was simply defined as its volume, as was done by Junqueira et al. (2013). In a similar way, the weight limit of each vehicle container is set as its volume, as well. In the case of the weight bearing strength, the computational experiments considered five different scenarios to analyze the impact of the constraints related to this factor. Each scenario is associated with an integer value (i.e. 2, 4, 6, 8 or 10), which is multiplied with the weight of the item. For example, for the scenario where the multiplier is 2, it means that each box will have a weight bearing strength of double its weight. We will refer to this value as the Weight Bearing Strength Multiplier (WBSM) from this point forward. The last consideration for these experiments is related to the packing strategy employed in the building of the blocks. As stated in Section 4.1.1 some studies suggest using a (Z, Y, X) scheme; others, a (X, Z, Y) scheme. We compare these two schemes in the computational experiments.

5.2. Results and analysis

To the best of our knowledge, no other heuristic procedure has included the set of operational constraints (loading and routing) considered simultaneously in this study, in the context of a VRPLC. However, for the purpose of comparison we striped the heuristic procedure from some of the additional operational objectives and constraints. This first comparison was performed by disregarding the balancing of the vehicle fleet and the improvement of weight distribution. The results of the proposed hybrid heuristic are compared with those of the study by Bortfeldt and Homberger (2013), which includes the prior results from Moura and Oliveira (2009). A second comparison is done by using the results of the proposed heuristic for Stages 1 and 2 (which represent a partial feasible solution for the problem), with the results obtained after adding the operational conditions considered in Stage 3. In addition, this comparison also shows how the solution obtained from Stages 1 and 2 is improved.

5.2.1. Comparison with previous heuristic approaches

The results of the proposed hybrid heuristic are compared with those obtained from the GRASP based heuristic presented by Moura and Oliveira (2009), and the TS + Evolutionary Algorithm hybrid proposed by Bortfeldt and Homberger (2013). Tables 1 to 4 show this comparison in terms of the required vehicles and total distance traveled for all four instance classes. Each entry in the table is expressed as a pair $(v; d)$, where v is the required number of vehicles and d is the total traveled distance. The values for the columns 3 and 4 were taken from the work by Bortfeldt and Homberger (2013). In this regard, the values for the study by Bortfeldt and Homberger (2013) reflect the better result of their P1R2 heuristic with 3 or 4 phases. This is represented as $(v; d)_3$ or $(v; d)_4$ to indicate which configuration reported a better result. In case there is no subscript, it means that the results of both configurations were the same. The remaining columns show the results of the proposed heuristic, considering the different combinations of the WBSM and packing strategies (X, Z, Y) and (Z, Y, X). Clearly, the results from the P1R2 heuristic proposed by Bortfeldt and Homberger (2013) outperform the other results. It is also clear that, for the proposed heuristic, the packing strategy (X, Z, Y) provides better results than using the other packing strategy. When compared with the results from Moura and Oliveira (2009), the proposed heuristic provides competitive results, in terms of the total distance, as the WBSM increases. Given that the study by Moura and Oliveira (2009) did not consider the weight bearing strength of the boxes, that would explain the difference in the number of required vehicles with low WBSM values. However, it can be observed that the number of vehicles in the proposed heuristic gets closer to the value obtained by Moura and Oliveira (2009), and at times outperforms it, as the WBSM increases. Another reason for the difference in the results could be attributed to the way the items' blocks are built, as these are constructed with the aim of facilitating the operational constraints related to the balance of the vehicle fleet and the distribution of weight inside the container. All in all, the proposed heuristic is capable of obtaining competitive results for the tested instances.

Table 1
Results for instance class GI_I1

Instance	Moura & Oliveira	Bortfeldt & Homberger	Proposed hybrid heuristic									
			Corner sort XYZ					Corner sort ZYX				
			WBSM=2	WBSM=4	WBSM=6	WBSM=8	WBSM=10	WBSM=2	WBSM=4	WBSM=6	WBSM=8	WBSM=10
GI_I1_01	(9;762.59)	(8;625.03)	(9; 764.48)	(7; 783.96)	(7; 766.01)	(7; 815.42)	(6; 843.62)	(21; 1090.53)	(20; 1181.86)	(21; 1167.54)	(21; 1167.54)	(21; 1102.71)
GI_I1_02	(8;675.24)	(7;575.09)	(9; 903.33)	(7; 794.9)	(7; 793.52)	(7; 951.96)	(6; 822.33)	(21; 1117.1)	(20; 1158.31)	(21; 1112.97)	(21; 1112.97)	(21; 1154.94)
GI_I1_03	(6;1250.86)	(5;501.73)	(9; 849.85)	(7; 794.71)	(7; 783.42)	(7; 794.88)	(6; 844.2)	(21; 1119.57)	(20; 1175.6)	(21; 1102.71)	(21; 1102.71)	(21; 1102.71)
GI_I1_04	(6;605.72)	(4;557.5)	(10; 794.67)	(7; 794.16)	(7; 908.84)	(7; 917.62)	(6; 801.97)	(21; 1118.23)	(20; 1128.25)	(21; 1102.71)	(21; 1102.71)	(21; 1102.71)
GI_I1_05	(9;1398.47)	(6;550.83)	(9; 763.04)	(7; 740.87)	(7; 752.55)	(7; 684.67)	(6; 843.62)	(21; 1119.57)	(20; 1192.39)	(21; 1146.47)	(21; 1146.47)	(21; 1102.71)
GI_I1_06	(7;757.08)	(5;549.13)	(9; 728.52)	(7; 825.28)	(7; 817.07)	(7; 742.29)	(6; 859.36)	(21; 1118.23)	(20; 1141.13)	(21; 1102.71)	(21; 1102.71)	(21; 1102.71)
GI_I1_07	(6;1108.67)	(5;474.8)	(9; 766.25)	(7; 793.22)	(7; 786.11)	(7; 722.15)	(6; 865.09)	(21; 1118.23)	(20; 1155.51)	(21; 1102.71)	(21; 1102.71)	(21; 1102.71)
GI_I1_08	(5;397.19)	(4;576.52)$_4$	(10; 844.27)	(7; 802.06)	(7; 862.07)	(7; 825.18)	(6; 801.97)	(21; 1118.23)	(20; 1128.25)	(21; 1102.71)	(21; 1102.71)	(21; 1102.71)
GI_I1_09	(6;1050.7)	(5;517.29)	(9; 689.13)	(7; 819.3)	(7; 797.12)	(7; 782.41)	(6; 795.59)	(21; 1118.23)	(20; 1120.25)	(21; 1102.71)	(21; 1102.71)	(21; 1102.71)
GI_I1_10	(6;578.36)	(5;505.46)	(10; 835.67)	(7; 731.64)	(7; 716.54)	(7; 796.93)	(6; 736.59)	(21; 1118.23)	(20; 1120.25)	(21; 1102.71)	(21; 1102.71)	(21; 1102.71)
GI_I1_11	(6;1128.55)	(5;500.33)	(9; 874.75)	(7; 722.17)	(7; 924.73)	(7; 808.23)	(7; 753.84)	(21; 1095.91)	(20; 1141.13)	(21; 1102.71)	(21; 1102.71)	(21; 1102.71)
GI_I1_12	(5;980.97)	(4;557.9)	(9; 820.71)	(7; 788.8)	(7; 754.88)	(7; 771.33)	(6; 763.36)	(21; 1118.23)	(20; 1120.25)	(21; 1102.71)	(21; 1102.71)	(21; 1102.71)

Table 2
Results for instance class GI_I2

Instance	Moura & Oliveira	Bortfeldt & Homberger	Proposed hybrid heuristic									
			Corner sort XYZ					Corner sort ZYX				
			WBSM=2	WBSM=4	WBSM=6	WBSM=8	WBSM=10	WBSM=2	WBSM=4	WBSM=6	WBSM=8	WBSM=10
GI_I2_01	(5;2668.55)	(4;765.83)$_4$	(9; 732.19)	(7; 748.89)	(7; 728.59)	(7; 717.87)	(6; 709.55)	(21; 1118.23)	(20; 1094.99)	(21; 1102.71)	(21; 1102.71)	(21; 1102.71)
GI_I2_02	(5;2555.26)	(4;561.11)$_4$	(9; 720.19)	(7; 743.93)	(7; 725)	(7; 725)	(6; 743)	(21; 1118.23)	(20; 1130.71)	(21; 1102.71)	(21; 1102.71)	(21; 1102.71)
GI_I2_03	(5;2526.11)	(4;508.65)	(9; 764.68)	(7; 713.6)	(7; 713.6)	(7; 713.6)	(6; 742.48)	(21; 1118.23)	(20; 1130.71)	(21; 1102.71)	(21; 1102.71)	(21; 1102.71)
GI_I2_04	(5;1953.67)	(4;471.17)$_4$	(10; 778.72)	(7; 758.22)	(7; 733.04)	(7; 743.03)	(6; 729)	(21; 1118.23)	(20; 1094.99)	(21; 1102.71)	(21; 1102.71)	(21; 1102.71)
GI_I2_05	(5;635.96)	(4;529.95)$_4$	(10; 758.78)	(7; 739.51)	(7; 738.93)	(7; 699.09)	(6; 693.9)	(21; 1118.23)	(20; 1101.23)	(21; 1102.71)	(21; 1102.71)	(21; 1102.71)
GI_I2_06	(5;2394.25)	(4;542.6)$_4$	(9; 738.19)	(7; 745.4)	(7; 725)	(7; 725)	(6; 772.95)	(21; 1118.23)	(20; 1101.23)	(21; 1102.71)	(21; 1102.71)	(21; 1102.71)
GI_I2_07	(5;2187.27)	(4;470.82)$_4$	(9; 762.71)	(7; 713.6)	(7; 713.6)	(7; 713.6)	(6; 742.48)	(21; 1118.23)	(20; 1130.71)	(21; 1102.71)	(21; 1102.71)	(21; 1102.71)
GI_I2_08	(4;472.35)	(4;465.13)$_4$	(10; 778.72)	(7; 742.73)	(7; 694.25)	(7; 743.03)	(6; 734.5)	(21; 1118.23)	(20; 1094.99)	(21; 1102.71)	(21; 1102.71)	(21; 1102.71)
GI_I2_09	(5;674.01)	(4;442.3)$_4$	(10; 831.93)	(7; 759.8)	(7; 731.04)	(7; 759.51)	(6; 721.76)	(21; 1118.23)	(20; 1120.25)	(21; 1102.71)	(21; 1102.71)	(21; 1102.71)
GI_I2_10	(5;753.04)	(4;578.13)	(9; 731.11)	(7; 762.66)	(7; 724.66)	(7; 724.22)	(7; 711.7)	(21; 1118.23)	(20; 1101.23)	(21; 1102.71)	(21; 1102.71)	(21; 1102.71)
GI_I2_11	(5;2049.39)	(4;477.16)$_4$	(9; 724.44)	(7; 740.85)	(7; 740.85)	(7; 707.09)	(6; 713.68)	(21; 1118.23)	(20; 1120.25)	(21; 1102.71)	(21; 1102.71)	(21; 1102.71)

Table 3

Results for instance class GII_I1

Instance	Moura & Oliveira	Bortfeldt & Homberger	Proposed hybrid heuristic									
			Corner sort XYZ					Corner sort ZYX				
			WBSM=2	WBSM=4	WBSM=6	WBSM=8	WBSM=10	WBSM=2	WBSM=4	WBSM=6	WBSM=8	WBSM=10
GII_I1_01	(9;823.04)	(8;654.62)	(13; 929.63)	(10; 788.58)	(9; 832.36)	(8; 760.52)	(8; 801.56)	(20; 1124.85)	(22; 1163.84)	(22; 1178.23)	(22; 1178.23)	(22; 1189)
GII_I1_02	(9;1622.59)	(7;592.14)	(13; 873.72)	(10; 917.68)	(9; 818.46)	(8; 834.78)	(8; 823.75)	(20; 1109.01)	(22; 1147.68)	(22; 1179.08)	(22; 1179.08)	(22; 1174.97)
GII_I1_03	(7;1451.39)	(6;548.49)	(13; 886.06)	(10; 777.19)	(9; 794.06)	(9; 929.34)	(9; 929.82)	(20; 1098.68)	(22; 1153.19)	(22; 1178.23)	(22; 1178.23)	(22; 1158.23)
GII_I1_04	(7;1221.44)	(6;540.43)$_4$	(13; 825.63)	(10; 842.38)	(9; 836.99)	(8; 821.4)	(8; 781.01)	(20; 1098.68)	(22; 1153.19)	(22; 1178.23)	(22; 1178.23)	(22; 1158.23)
GII_I1_05	(10;1532.44)	(6;693.22)	(13; 848.93)	(10; 881.23)	(9; 846.97)	(8; 834.52)	(8; 860.51)	(20; 1101.43)	(22; 1153.19)	(22; 1144.82)	(22; 1144.82)	(22; 1178.23)
GII_I1_06	(8;1576.1)	(6;627.53)	(13; 848.93)	(10; 812.22)	(9; 853.15)	(8; 907.47)	(8; 721.15)	(20; 1085.76)	(22; 1144.82)	(22; 1144.82)	(22; 1144.82)	(22; 1158.23)
GII_I1_07	(7;1378.36)	(6;622.5)	(13; 829.37)	(10; 850.61)	(9; 802.41)	(9; 800.42)	(9; 781.31)	(20; 1091.64)	(22; 1144.82)	(22; 1144.82)	(22; 1144.82)	(22; 1158.23)
GII_I1_08	(7;1187.52)	(6;549.18)$_4$	(13; 849.66)	(10; 808.96)	(9; 822.43)	(8; 817.49)	(8; 845.03)	(20; 1091.64)	(22; 1144.82)	(22; 1144.82)	(22; 1144.82)	(22; 1158.23)
GII_I1_09	(6;625.91)	(6;709.61)	(13; 879.01)	(10; 780.24)	(9; 805.66)	(8; 772.11)	(8; 743.05)	(20; 1082.34)	(22; 1144.82)	(22; 1144.82)	(22; 1144.82)	(22; 1158.23)
GII_I1_10	(7;1235.62)	(6;570.38)$_4$	(13; 885.34)	(10; 851.28)	(9; 781.77)	(8; 797.34)	(8; 819.43)	(20; 1092.27)	(22; 1144.82)	(22; 1144.82)	(22; 1144.82)	(22; 1158.23)
GII_I1_11	(7;1293.95)	(6;540.03)	(13; 839.22)	(10; 813.76)	(9; 828.78)	(8; 900.8)	(8; 900.8)	(20; 1091.64)	(22; 1144.82)	(22; 1144.82)	(22; 1144.82)	(22; 1158.23)
GII_I1_12	(7;1069.11)	(6;539.84)$_4$	(13; 901.42)	(10; 817.49)	(9; 829.67)	(8; 719.46)	(8; 719.46)	(20; 1088.02)	(22; 1144.82)	(22; 1144.82)	(22; 1144.82)	(22; 1158.23)

Table 4

Results for instance class GII_I2

Instance	Moura & Oliveira	Bortfeldt & Homberger	Proposed hybrid heuristic									
			Corner sort XYZ					Corner sort ZYX				
			WBSM=2	WBSM=4	WBSM=6	WBSM=8	WBSM=10	WBSM=2	WBSM=4	WBSM=6	WBSM=8	WBSM=10
GII_I2_01	(7;3740.55)	(6;591.35)	(13; 849.47)	(10; 735.79)	(10; 770.04)	(8; 780.09)	(8; 770.3)	(20; 1088.02)	(22; 1144.82)	(22; 1144.82)	(22; 1144.82)	(22; 1158.23)
GII_I2_02	(7;3496.39)	(6;560.88)	(13; 869.48)	(10; 823.92)	(9; 809.06)	(8; 751.4)	(8; 754.98)	(20; 1091.64)	(22; 1144.82)	(22; 1144.82)	(22; 1144.82)	(22; 1158.23)
GII_I2_03	(7;3134.62)	(6;525.99)$_4$	(13; 849.47)	(10; 798.56)	(9; 770.31)	(9; 774.97)	(9; 770.05)	(20; 1091.64)	(22; 1144.82)	(22; 1144.82)	(22; 1144.82)	(22; 1158.23)
GII_I2_04	(6;3814.29)	(6;516.35)	(13; 875.67)	(10; 774.56)	(9; 817.85)	(8; 778.87)	(8; 778.87)	(20; 1091.64)	(22; 1144.82)	(22; 1144.82)	(22; 1144.82)	(22; 1158.23)
GII_I2_05	(7;627.66)	(6;649.8)$_4$	(13; 849.47)	(10; 735.79)	(10; 770.04)	(8; 769)	(8; 769)	(20; 1088.02)	(22; 1144.82)	(22; 1144.82)	(22; 1144.82)	(22; 1158.23)
GII_I2_06	(7;3115.18)	(6;560.64)$_4$	(13; 872.97)	(10; 790.09)	(9; 804.45)	(8; 719.4)	(8; 782.36)	(20; 1091.64)	(22; 1144.82)	(22; 1144.82)	(22; 1144.82)	(22; 1158.23)
GII_I2_07	(7;2740.03)	(6;506.16)$_4$	(13; 829.37)	(10; 798.56)	(9; 770.31)	(9; 774.97)	(9; 770.05)	(20; 1079.11)	(22; 1144.82)	(22; 1144.82)	(22; 1144.82)	(22; 1158.23)
GII_I2_08	(7;2212.02)	(6;534.97)$_4$	(13; 875.67)	(10; 774.56)	(9; 817.85)	(8; 778.87)	(8; 778.87)	(20; 1091.64)	(22; 1144.82)	(22; 1144.82)	(22; 1144.82)	(22; 1158.23)
GII_I2_09	(7;2962.35)	(6;544.63)$_4$	(13; 885.34)	(10; 773.29)	(9; 796.12)	(8; 703.37)	(8; 776.51)	(20; 1088.02)	(22; 1144.82)	(22; 1144.82)	(22; 1144.82)	(22; 1158.23)
GII_I2_10	(7;3512.25)	(6;599.93)	(13; 869.48)	(10; 778.22)	(9; 780.92)	(8; 748.59)	(8; 748.59)	(20; 1091.64)	(22; 1144.82)	(22; 1144.82)	(22; 1144.82)	(22; 1158.23)
GII_I2_11	(6;2631.39)	(6;501.26)$_4$	(13; 849.47)	(10; 765.05)	(9; 758.21)	(8; 784.72)	(8; 784.72)	(20; 1088.02)	(22; 1144.82)	(22; 1144.82)	(22; 1144.82)	(22; 1158.23)

5.2.2. Analysis of the impact of weight distribution and fleet balancing

Tables 5 to 8 provide the computational results produced by the hybrid heuristic. The configuration of each test is as follows. The column 'BWSM' shows the values used to test the impact of considering the weight bearing strength of the items to be delivered. The column 'Constraints' uses four values that represent the possible combinations by including or not including the conditions to balance the vehicle fleet (BF and NBF), and by including or not the weight distribution inside the container of the vehicle (WD and NWD). The values in the column 'Corner Sorting' assist to differentiate between the two strategies for sorting insertion corners in the packing stage of the hybrid heuristic.

In addition, the remaining columns in the tables present the computational time required to solve the corresponding instance problem (Time) and the number of vehicles (Routes) used to deliver all the items defined in the problem. The remaining columns present the values for the objective functions. Given the number of different instances and space limitations, this section provides the results for one problem from each class. For the purpose of replicating the experiments, the remaining results are available upon request.

Table 5
Results of the proposed heuristic for GI_I1_01 using additional constraints

WBSM	Constraints	Corner sorting XZY						Corner sorting ZYX					
		Time (secs)	Routes	z1	z2	z3	z4	Time (secs)	Routes	z1	z2	z3	z4
2	NBF&NWD	24.14	9	764.48	12.03	78920	230.47	51.4	21	1090.53	0	57360	961.07
	BF&NWD	8.32	11	1001.03	0	49520	237.79	33.53	21	1188.71	0	52680	874.88
	NBF&WD	8.54	9	899.59	2.83	57440	85.93	36.17	21	1183.36	0	53880	609.4
	BF&WD	8.7	11	997.8	0	49520	60.86	30.42	21	1188.71	0	52680	539.71
4	NBF&NWD	20.92	7	783.96	165.91	65600	240.83	48.83	20	1181.86	0	52680	759.82
	BF&NWD	8.45	8	884.97	0	49520	260.06	26.79	21	1192.02	0	53760	730.2
	NBF&WD	6.24	7	866.53	158.08	66680	95.03	28.06	20	1189.44	0	53400	584.63
	BF&WD	8.5	8	884.97	0	49520	114.91	28.02	21	1192.02	0	53760	508.28
6	NBF&NWD	23.74	7	766.01	12.03	78920	312.46	48.79	21	1167.54	0	53880	811.08
	BF&NWD	10.28	8	930.27	30.4	52400	306.98	27.79	21	1201.19	0	52680	725.12
	NBF&WD	7.89	7	852.67	2.83	57440	176.2	28.66	21	1197.02	0	53880	607.8
	BF&WD	11.96	8	902.86	30.4	52400	178.11	29.05	21	1201.19	0	52680	491.95
8	NBF&NWD	19.56	7	815.42	0	62000	371.91	48.31	21	1167.54	0	53880	829.44
	BF&NWD	9.34	8	890.7	0	49520	307.33	27.77	21	1201.19	0	52680	725.12
	NBF&WD	7.28	7	850.41	26.06	66680	198.65	28.41	21	1197.02	0	53880	599.46
	BF&WD	8.96	8	922.64	21.26	54320	220.6	29.07	21	1201.19	0	52680	502.78
10	NBF&NWD	19.69	6	843.62	12.03	78920	558.79	48.12	21	1102.71	0	55560	855.87
	BF&NWD	6.66	7	934.91	377.52	59000	401.42	28	21	1215.92	0	52680	771.4
	NBF&WD	5.62	6	833.45	146.27	69800	270.89	28.95	21	1197.02	0	53880	593.37
	BF&WD	7.16	7	934.91	377.52	59000	207.46	29.24	21	1215.92	0	52680	535.26

Table 6
Results of the proposed heuristic for GI_I2_01 using additional constraints

WBSM	Constraints	XZY						ZYX					
		Time (secs)	Routes	z1	z2	z3	z4	Time (secs)	Routes	z1	z2	z3	z4
2	NBF&NWD	32.1	9	732.19	0	78920	254.5	58.24	21	1118.23	0	57360	990.17
	BF&NWD	8.36	11	1014.01	0	49520	225.87	34.98	21	1188.71	0	52680	869.34
	NBF&WD	8.32	9	899.59	0	57440	85.18	35.61	21	1183.36	0	53880	591.89
	BF&WD	8.67	11	1014.01	0	49520	70.76	29.73	21	1188.71	0	52680	557.28
4	NBF&NWD	36.85	7	748.89	0	65360	271.85	48.09	20	1094.99	0	55560	852.4
	BF&NWD	7.5	8	931.25	0	49520	285.63	28.02	21	1192.02	0	53760	754.05
	NBF&WD	6.12	7	886.8	0	64160	119.76	27.68	20	1183.27	0	53400	544.81
	BF&WD	8.58	8	895.09	0	52040	93.62	27.03	21	1192.02	0	53760	508.44
6	NBF&NWD	36.19	7	728.59	0	71120	314.87	51.41	21	1102.71	0	55560	922.3
	BF&NWD	9.16	8	891.25	0	49520	280	29.04	21	1201.19	0	52680	728.65
	NBF&WD	7.76	7	855.45	0	57440	168.27	28.87	21	1197.02	0	53880	583.8
	BF&WD	9.8	8	908.86	0	53840	222.24	28.41	21	1201.19	0	52680	502.78
8	NBF&NWD	23.68	7	717.87	0	58040	340.8	51.15	21	1102.71	0	55560	898.3
	BF&NWD	9.05	8	938.31	0	54320	357.4	29.06	21	1201.19	0	52680	749.12
	NBF&WD	7.17	7	853.18	0	66680	142.67	28.56	21	1197.02	0	53880	601.33
	BF&WD	13.23	8	918.21	0	53600	237.85	28.13	21	1201.19	0	52680	514.07
10	NBF&NWD	35.92	6	709.55	0	78920	433.91	50.28	21	1102.71	0	55560	879.87
	BF&NWD	6.5	7	913.43	0	59000	389.22	29.45	21	1191.93	0	52680	747.4
	NBF&WD	5.65	6	843.93	0	69800	234.22	29.09	21	1197.02	0	53880	562.01
	BF&WD	7.85	7	913.43	0	59000	209.56	28.53	21	1191.93	0	52680	515.23

Table 7
Results of the proposed heuristic for GII_I1_01 using additional constraints

WBSM	Constraints	XZY Time (secs)	Routes	z1	z2	z3	z4	ZYX Time (secs)	Routes	z1	z2	z3	z4
2	NBF&NWD	34.19	13	929.63	0	74000	172.86	58.55	20	1124.85	0	77400	860.42
	BF&NWD	8.17	14	1049.79	0	74000	181.72	29.94	20	1175.16	0	74000	685.47
	NBF&WD	8.15	13	1049.16	0	74000	51.41	29.99	20	1168.86	0	76080	665.89
	BF&WD	7.95	14	1049.79	0	74000	47.44	30.3	20	1175.16	0	74000	556.91
4	NBF&NWD	27.3	10	788.58	0	77600	247.72	70.2	22	1163.84	0	79680	839.74
	BF&NWD	8.46	11	955.72	0	74000	273.1	30.66	23	1242.81	0	78960	754.72
	NBF&WD	6.93	10	933.26	0	77600	110.65	29.31	22	1185.67	0	77400	465
	BF&WD	7.93	11	939.11	0	74000	96.95	30.01	23	1242.81	0	78960	497.76
6	NBF&NWD	23.41	9	832.36	0	77600	178.94	56.19	22	1178.23	0	77400	793.54
	BF&NWD	9.11	9	947.37	0	74000	206.87	25.93	23	1242.81	0	78960	800.31
	NBF&WD	8.27	9	918.38	0	77600	87.92	25.58	22	1185.67	0	77400	521.51
	BF&WD	8.7	9	947.37	0	74000	115.3	25.49	23	1242.81	0	78960	562.78
8	NBF&NWD	28.41	8	760.52	0	77600	180.19	56.3	22	1178.23	0	77400	816.61
	BF&NWD	9.66	8	944.38	0	74000	233.08	25.85	23	1242.81	0	78960	834.83
	NBF&WD	8.98	8	894.72	0	77600	101.61	25.67	22	1185.67	0	77400	554.05
	BF&WD	9.48	8	886.65	0	74000	70.05	26.08	23	1242.81	0	78960	514.78
10	NBF&NWD	22.33	8	801.56	0	77600	362.77	55.62	22	1189	0	77400	770.23
	BF&NWD	7.07	9	926.02	0	74000	366.81	26.46	23	1242.81	0	78960	813.59
	NBF&WD	7.2	8	894.72	0	77600	188.72	26.38	22	1178.23	0	77400	491.71
	BF&WD	7.34	9	926.02	0	74000	163.35	26.57	23	1242.81	0	78960	509.25

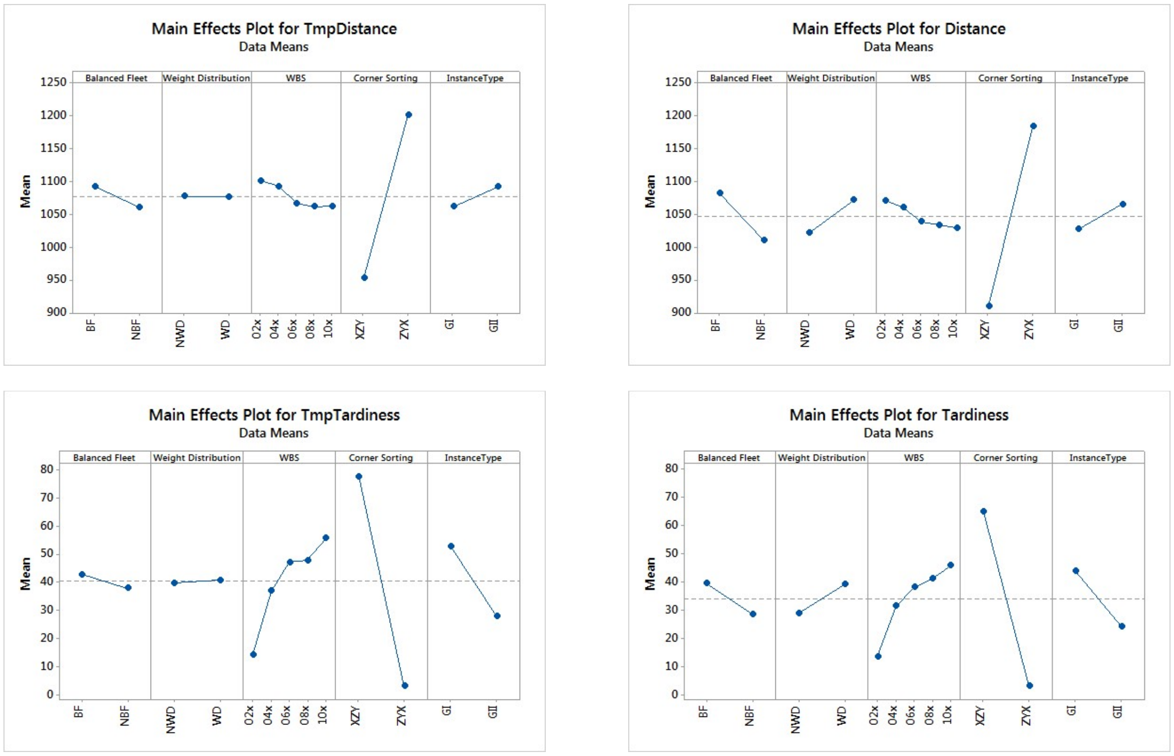
Table 8
Results of the proposed heuristic for GII_I2_01 using additional constraints

WBSM	Constraints	XZY						ZYX					
		Time (secs)	Routes	z1	z2	z3	z4	Time (secs)	Routes	z1	z2	z3	z4
2	NBF&NWD	36.47	13	849.47	0	74000	169.04	44.07	15	860.34	0	74000	838.93
	BF&NWD	8.2	14	1049.79	0	74000	170.98	19.13	15	1043.74	0	74000	953.04
	NBF&WD	8.25	13	1049.16	0	74000	50.21	18.84	15	1113.37	0	74000	685.07
	BF&WD	8.36	14	1049.79	0	74000	45.88	19.16	16	1086.21	0	74000	754.59
4	NBF&NWD	34.85	10	735.79	0	77600	244.26	44.56	12	834.24	0	74000	713.86
	BF&NWD	8.72	11	915.18	0	74000	280.95	20.56	15	1070.52	0	74000	648.28
	NBF&WD	6.93	10	937.87	0	77600	95.69	21.16	12	1037.85	0	74000	1040.65
	BF&WD	8.21	11	939.11	0	74000	110.57	20.86	14	1113.24	0	74000	747.31
6	NBF&NWD	29.18	10	770.04	0	80600	231.71	50.79	12	842.32	0	74000	842.12
	BF&NWD	8.77	9	984.75	0	74000	178.26	21.26	14	1059.79	0	74000	764.76
	NBF&WD	8.4	10	937.87	0	77600	84.1	21.44	12	971.51	0	74000	933.62
	BF&WD	8.99	9	984.75	0	74000	57.82	21.56	13	1063.93	0	74000	695.87
8	NBF&NWD	29.65	8	780.09	0	77600	196.31	50.2	12	822.96	0	74000	777.01
	BF&NWD	11.48	9	936.13	0	74000	178.44	21.41	12	998.49	0	74000	761.19
	NBF&WD	9.01	8	904.51	0	77600	100.39	22.12	12	1003.7	0	74000	985.76
	BF&WD	9.6	8	929.3	0	74000	71.24	21.54	12	998.49	0	74000	731.1
10	NBF&NWD	28.31	8	770.3	0	77600	348.9	53	12	851.37	0	74000	762
	BF&NWD	7.9	9	897.04	0	74000	374.55	21.7	13	986.19	0	74000	693.69
	NBF&WD	7.07	8	904.51	0	77600	161.68	22.04	12	1013.82	0	74000	738.77
	BF&WD	7.64	9	897.04	0	74000	173.79	21.99	13	986.19	0	74000	683.69

In terms of the required computational time, the hybrid heuristic was able to solve most cases in less than 30 seconds when using packing strategy (X, Z, Y). It is interesting to see, that breaching the 30 seconds mark occurs when sorting the insertion corners under a (Z, Y, X) strategy. The increments in the processing time may be because this strategy produces blocks with larger surface areas and, as the procedure for loading the vehicles of the hybrid heuristic avoids placing weight on top of the generated blocks, this results in a considerable increment in the number of routes and, hence, in a larger computational effort in the local search procedure of Stage 3. As opposed to privileging the insertion corners located in the lower positions first, the (X, Z, Y) sorting tries to stack items early on during the block construction process, which results in fewer routes to deliver all the items to all the clients, and in better use of the container's available storage space. Another aspect that influences the total number of routes is of course the weight bearing constraints. Once again, the number of routes decreases as the WBSM increases.

Finally, an analysis of variance (ANOVA) with a significance of 5%, was carried out for each one of the four response variables (z_1, z_2, z_3, z_4), to better understand the impact of the consideration of the additional constraints in Stage 3 of the proposed hybrid heuristic, on the objective values after Stages 1 and 2. Five factors were considered: (i) if whether balancing the vehicle fleet was considered, or not (i.e. BF or NBF); (ii) if whether weight distribution was improved, or not (i.e. WD or NWD); (iii) the value of the WBSM (i.e. 2, 4, 6, 8 and 10); (iv) the packing strategy (i.e. (X, Z, Y) or (Z, Y, X)); and (v) the instance type (i.e. GI or GII). As such, there were a total of 80 treatments and 80 x 23 observations for each ANOVA, and all factors report a significant effect in the four response variables.

Fig. 13 presents the main effects plots for the considered factors in the ANOVAs. In this figure, graphs with the label "Tmp" (which are always on the left) refer to the results obtained after Stages 1 and 2; "ML -ml", to $maxvload - minvload$ for the third objective function; and "COG-Y", to the total deviation of the COG inside the container along the width of the container. Note that for all cases, the results from Stages 1 and 2 are improved in Stage 3. Furthermore, this graphical analysis shows that the weight bearing strength constraints and the packing strategy have a significant impact on the value of the objective functions, indicating that a high WBSM and the packing strategy (X, Z, Y) provide better results. However, the opposite happens with the tardiness as a low value of WBSM and the use of the (Z, Y, X) strategy provides a lower tardiness. This is not surprising, as these configurations generate more delivery routes. Thus, only a few vehicles would have to visit several clients, which in turn would reduce the possibility of late deliveries and any penalties. However, in a real-life scenario this could result in overhead costs due to the maintenance of a large vehicle fleet and the low use of container space. Lastly, depending on the type of instance, different values for the objective functions are obtained as well. Values for z_1 and z_3 were better in instances of type GI, while values for z_2 and z_4 were better in instances of type GII.



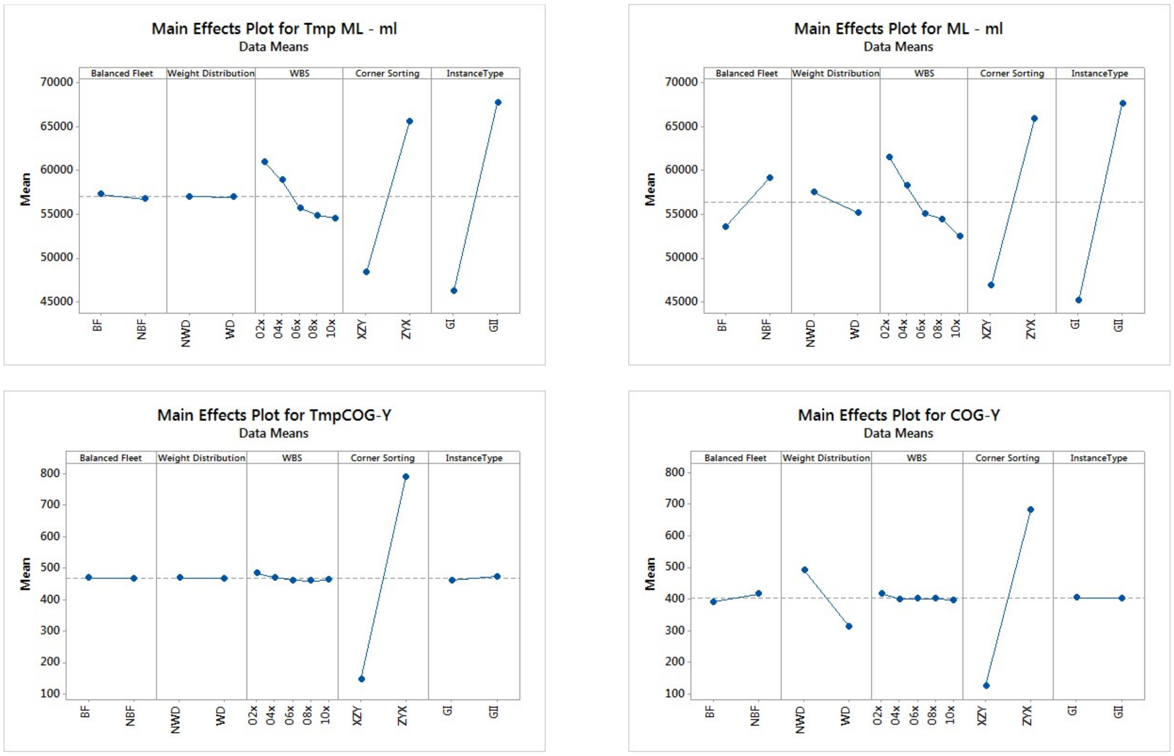


Fig. 13. Impact of additional constraints with different testing configurations

6. Conclusions and future work

The VRPLC is related to real life transportation situations and has received increased attention in recent years. In this regard, a hybrid heuristic solution method based on the "pack first – route second" approach proposed by Bortfeldt and Homberger (2013), has been proposed to solve a version of the VRPLC which considers several practical loading and routing constraints. Among such constraints are the vertical stability of cargo, the load bearing strength of items, the weight limit of the container, the unloading sequence, the weight's distribution, delivery time windows and a balanced vehicle fleet. Consequently, these constraints add to the complexity of the VRPLC, which motivated the development of a hybrid heuristic procedure than can be easily implemented by operational researchers and distribution managers. The proposed hybrid solution is also capable of integrating simultaneously a set of complex operational aspects and of generating feasible solutions in a short amount of time. With these considerations, the method may serve as a useful tool in both academic research and real-life applications in distribution logistics.

The proposed hybrid heuristic which consists of three stages, expands the procedure presented by Vega-Mejía and Montoya-Torres (2017). The first stage consists of a GRASP metaheuristic that finds a suitable packing arrangement for each client. The second stage combines a CWS algorithm and the GRASP metaheuristic to determine the delivery routes and cargo pattern of each vehicle. The final stage performs a local search to balance the vehicle fleet and to improve the weight distribution inside each vehicle. To test the method's performance, computational experiments were performed under different configurations. The numeric results show that sorting the insertion corners by their (X, Z, Y) coordinates during the packing stage, outperforms results that employ the (Z, Y, X) approach. Additional comparisons were carried out to analyze the impact of different operational constraints. On one hand, both the inclusion of the weight's distribution and balancing the vehicle fleet show improvements when compared to the cases that did not consider these aspects. On the other hand, the results obtained when including a low weight bearing strength for the carried items are outperformed as the load bearing index (i.e. fragility) increases.

It is arguable then that, even though finding a smaller number of vehicles can be achieved without considering the fragility of the cargo, the impact of overlooking this factor may carry with it the generation of solutions that might not be feasible in real applications. This is consistent with the argument presented by Bischoff and Ratcliff (1995), that not considering additional constraints may result in impractical solutions. However, rather than a limitation, this creates an interesting opportunity to extend future research into the development of solution methods for more complex VRPLCs. For instance, and with regard to the constraints designed to address the weight resistance of either the items or the container, these constraints could be softened as suggested by Zachariadis et al. (2016) to allow a greater variety of solutions from other heuristic procedures and to analyze their performance.

There are other directions that we consider interesting, and which could further improve future heuristic applications, for example the inclusion of other operational constraints (e.g. pick up & delivery, partial deliveries). In addition, given that multiple operational constraints may result in multiple objectives, often conflicting with one another, methods should be designed to handle these multiple objectives from the perspective of non-dominated solutions, as suggested by Montoya-Torres et al. (2015). Since the use of the ‘air-space’ inside each vehicle container is critical, enhancement of the placement strategies during the packing procedures still deserves attention. Moreover, the number of required vehicles could be improved by considering the generation of blocks that can interlock with others. This would represent an interesting technical challenge, as the consideration of swapping blocks between vehicles to balance the vehicle fleet and the reflection or rotation of blocks to improve the COG could result in the destruction of cargo patterns and increase computational time. Regarding computation of the COG of the loaded vehicle, the proposed heuristic could be improved by considering metrics from vehicle dynamics, such as the Lateral Transfer Ratio to drive the distribution of weight inside containers towards more realistic scenarios (see Ramos, Silva, & Oliveira, 2018). Another way to reduce the number of required vehicles could be by using vehicles with multiple stacking levels, as employed by Iori and Riera-Ledesma (2015). These stacking levels would provide efficient vertical support at different heights inside the container. Some difficulties may arise, such as the generation of several blocks for a single client and the additional constraints that consider the height of the stacks. Finally, other directions for further research worth exploring could be the realignment of loading and routing operations to a triple-bottom-line approach (i.e. economic, environmental and societal factors), as expressed by Montoya-Torres et al. (2015) and Vega-Mejía et al. (2019b), for sustainable transportation.

References

Alinaghian, M., Zamanlou, K., & Sabbagh, M. S. (2017). A bi-objective mathematical model for two-dimensional loading time-dependent vehicle routing problem. *Journal of the Operational Research Society*, *68*(11), 1422–1441.

Baldi, M. M., Perboli, G., & Tadei, R. (2012). The three-dimensional knapsack problem with balancing constraints. *Applied Mathematics and Computation*, *218*(19), 9802–9818.

Bischoff, E. E., & Ratcliff, M. S. W. (1995). Issues in the development of approaches to container loading. *Omega*, *23*(4), 377–390.

Bortfeldt, A., Hahn, T., Männel, D., & Mönch, L. (2015). Hybrid algorithms for the vehicle routing problem with clustered backhauls and 3D loading constraints. *European Journal of Operational Research*, *243*(1), 82–96.

Bortfeldt, A., & Homberger, J. (2013). Packing first, routing second — A heuristic for the vehicle routing and loading problem. *Computers & Operations Research*, *40*(3), 873–885.

Bortfeldt, A., & Wäscher, G. (2013). Constraints in container loading - A state-of-the-art review. *European Journal of Operational Research*, *229*(1), 1–20.

Caceres-Cruz, J., Arias, P., Guimarans, D., Riera, D., & Juan, A. A. (2014). Rich vehicle routing problem. *ACM Computing Surveys*, *47*(2), 1–28.

Ceschia, S., Schaerf, A., Stützle, T., & Stuzle, T. (2013). Local search techniques for a routing-packing problem. *Computers & Industrial Engineering*, *66*(4), 1138–1149.

Christofides, N., & Whitlock, C. (1977). An algorithm for two-dimensional cutting problems. *Operations Research*, *25*(1), 30–44.

Cordeau, J.-F., Gendreau, M., Laporte, G., Potvin, J.-Y., & Semet, F. (2002). A guide to vehicle routing heuristics. *Journal of the Operational Research Society*, *53*(5), 512–522.

Cui, Y. (2007). Exact algorithm for generating two-segment cutting patterns of punched strips. *Applied Mathematical Modelling*, *31*(9), 1865–1873.

Davies, A. P., & Bischoff, E. E. (1999). Weight distribution considerations in container loading. *European Journal of Operational Research*, *114*(3), 509–527.

Dominguez, O., Guimarans, D., Juan, A. A., & de la Nuez, I. (2016a). A biased-randomised large neighbourhood search for the two-dimensional vehicle routing problem with backhauls. *European Journal of Operational Research*, *255*(2), 442–462.

Dominguez, O., Juan, A. A., Barrios, B., Faulin, J., & Agustin, A. (2016b). Using biased randomization for solving the two-dimensional loading vehicle routing problem with heterogeneous fleet. *Annals of Operations Research*, *236*(2), 383–404.

Dominguez, O., Juan, A. A., de la Nuez, I., & Ouelhadj, D. (2016c). An ILS-biased randomization algorithm for the two-dimensional loading HFVRP with sequential loading and items rotation. *Journal of the Operational Research Society*, *67*, 37–53.

Dominguez, O., Juan, A. A., & Faulin, J. (2014). A biased-randomized algorithm for the two-dimensional vehicle routing problem with and without item rotations. *International Transactions in Operational Research*, *21*(3), 375–398.

Egeblad, J., Garavelli, C., Lisi, S., & Pisinger, D. (2010). Heuristics for container loading of furniture. *European Journal of Operational Research*, *200*(3), 881–892.

Eley, M. (2002). Solving container loading problems by block arrangement. *European Journal of Operational Research*, *141*(2), 393–409.

Fuellerer, G., Doerner, K. F., Hartl, R. F., & Iori, M. (2010). Metaheuristics for vehicle routing problems with three-dimensional loading constraints. *European Journal of Operational Research*, *201*(3), 751–759.

García-Cáceres, R. G., Vega-Mejía, C. A., & Caballero-Villalobos, J. P. (2011). Integral optimization of the container loading problem. In I. Dritsas (Ed.), *Stochastic Optimization - Seeing the Optimal for the Uncertain* (1st ed., pp. 225–254). Rijeka, Croatia: InTech.

Gendreau, M., Iori, M., Laporte, G., & Martello, S. (2006). A tabu search algorithm for a routing and container loading problem. *Transportation Science*, *40*(3), 342–350.

Guimarans, D., Dominguez, O., Juan, A. A., & Martinez, E. (2016). A multi-start simheuristic for the stochastic two-dimensional vehicle routing problem. In T. Roede, P. Frazier, R. Szechtman, & E. Zhou (Eds.), *2016 Winter Simulation Conference (WSC)* (pp. 2326–2334). Washington, US: IEEE Publishing.

Guimarans, D., Dominguez, O., Panadero, J., & Juan, A. A. (2018). A simheuristic approach for the two-dimensional vehicle routing problem with stochastic travel times. *Simulation Modelling Practice and Theory*, *89*(May), 1–14.

Hokama, P., Miyazawa, F. K., & Xavier, E. C. (2016). A branch-and-cut approach for the vehicle routing problem with loading constraints. *Expert Systems with Applications*, *47*, 1–13.

Iori, M., & Martello, S. (2010). Routing problems with loading constraints. *Top*, *18*(1), 4–27.

Iori, M., & Riera-Ledesma, J. (2015). Exact algorithms for the double vehicle routing problem with multiple stacks. *Computers & Operations Research*, *63*, 83–101.

Juan, A. A., Faulin, J., Grasman, S. E., Rabe, M., & Figueira, G. (2015). A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. *Operations Research Perspectives*, *2*, 62–72.

Junqueira, L., & Morabito, R. (2015). Heuristic algorithms for a three-dimensional loading capacitated vehicle routing problem in a carrier. *Computers & Industrial Engineering*, *88*, 110–130.

Junqueira, L., Oliveira, J. F., Carravilla, M. A., & Morabito, R. (2013). An optimization model for the vehicle routing problem with practical three-dimensional loading constraints. *International Transactions in Operational Research*, *20*(5), 645–666.

Koch, H., Bortfeldt, A., & Wäscher, G. (2018). A hybrid algorithm for the vehicle routing problem with backhauls, time windows and three-dimensional loading constraints. *OR Spectrum*, *40*, 1029–1075.

Kramer, R., Subramanian, A., Vidal, T., & Cabral, L. dos A. F. (2015). A matheuristic approach for the pollution-routing problem. *European Journal of Operational Research*, *243*(2), 523–539.

Laporte, G. (2009). Fifty years of vehicle routing. *Transportation Science*, *43*(4), 408–416.

Lim, A., Ma, H., Qiu, C., & Zhu, W. (2013). The single container loading problem with axle weight constraints. *International Journal of Production Economics*, *144*(1), 358–369.

Lin, C., Choy, K. L., Ho, G. T. S., Chung, S. H., & Lam, H. Y. (2014). Survey of green vehicle routing problem: Past and future trends. *Expert Systems with Applications*, *41*(4), 1118–1138.

Mehrjerdi, Y. Z. (2014). A multiple objective stochastic approach to vehicle routing problem. *International Journal of Advanced Manufacturing Technology*, *74*(5–8), 1149–1158.

Montoya-Torres, J. R., López Franco, J., Nieto Isaza, S., Felizzola Jiménez, H., & Herazo-Padilla, N. (2015). A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering*, *79*, 115–129.

Moura, A., & Oliveira, J. F. (2009). An integrated approach to the vehicle routing and container loading problems. *Or Spectrum*, *31*(4), 775–800.

Pisinger, D. (2002). Heuristics for the container loading problem. *European Journal of Operational Research*, *141*(2), 382–392.

Ramos, A. G., Silva, E., & Oliveira, J. F. (2018). A new load balance methodology for container loading problem in road transportation. *European Journal of Operational Research*, *266*(3), 1140–1152.

Resende, M. G. C., & Ribeiro, C. C. (2010). Greedy randomized adaptive search procedures: Advances, hybridizations, and applications. In M. Gendreau & J.-Y. Potvin (Eds.), *Handbook of Metaheuristics* (2nd ed., pp. 283–319). New York, NY: Springer New York.

Sicilia, J. A., Quemada, C., Royo, B., & Escuin, D. (2016). An optimization algorithm for solving the rich vehicle routing problem based on variable neighborhood search and tabu search metaheuristics. *Journal of Computational and Applied Mathematics*, *291*, 468–477.

Tricoire, F., Doerner, K. F., Hartl, R. F., & Iori, M. (2011). Heuristic and exact algorithms for the multi-pile vehicle routing problem. *Or Spectrum*, *33*(4), 931–959.

Vega-Mejía, C. A., & Montoya-Torres, J. R. (2017). A Hybrid Heuristic for a Balanced Vehicle Routing Problem with Time Windows and Loading Constraints. In A. Duarte, A. Viana, A. A. Juan, B. Mélian, & H. Ramalhinho (Eds.), *MIC 2017 - 12th Metaheuristics International Conference* (pp. 617–626). Barcelona, Es: Universitat Pompeu Fabra. Retrieved from http://mic2017.upf.edu/proceedings/

Vega-Mejía, C. A., Montoya-Torres, J. R., & Islam, S. M. N. (2019a). A nonlinear optimization model for the balanced vehicle routing problem with loading constraints. *International Transactions in Operational Research*, *26*(3), 794–835.

Vega-Mejía, C. A., Montoya-Torres, J. R., & Islam, S. M. N. (2019b). Consideration of triple bottom line objectives for sustainability in the optimization of vehicle routing and loading operations: a systematic literature review. *Annals of Operations Research*, *273*(Operations Research for Transportation), 311–375.

Wäscher, G., Haußner, H., & Schumann, H. (2007). An improved typology of cutting and packing problems. *European Journal of Operational Research*, *183*(3), 1109–1130.

Zachariadis, E. E., Tarantilis, C. D., & Kiranoudis, C. T. (2013). Designing vehicle routes for a mix of different request types, under time windows and loading constraints. *European Journal of Operational Research*, *229*(2), 303–317.

Zachariadis, E. E., Tarantilis, C. D., & Kiranoudis, C. T. (2016). The vehicle routing problem with simultaneous pick-ups and deliveries and two-dimensional loading constraints. *European Journal of Operational Research*, *251*(2), 369–386.

Zhang, D., Cai, S., Ye, F., Si, Y.-W., & Nguyen, T. T. (2017). A hybrid algorithm for a vehicle routing problem with realistic constraints. *Information Sciences*, *394*, 167–182.

Zhang, Z., Wei, L., & Lim, A. (2015). An evolutionary local search for the capacitated vehicle routing problem minimizing fuel consumption under three-dimensional loading constraints. *Transportation Research Part B-Methodological*, *82*, 20–35.