Computers & Industrial Engineering 88 (2015) 110–130

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Heuristic algorithms for a three-dimensional loading capacitated vehicle routing problem in a carrier ☆

Leonardo Junqueira [a], Reinaldo Morabito [b,*]

[a] Programa de Pós-Graduação em Engenharia de Produção, Universidade Nove de Julho, Avenida Francisco Matarazzo, 612, 05001-100 Água Branca, São Paulo, São Paulo, Brazil
[b] Departamento de Engenharia de Produção, Universidade Federal de São Carlos, Rodovia Washington Luís, km 235 – SP-310, 13565-905 São Carlos, São Paulo, Brazil

A R T I C L E I N F O

Article history:
Received 22 August 2014
Received in revised form 10 March 2015
Accepted 9 June 2015
Available online 18 June 2015

Keywords:
Capacitated vehicle routing problem
Three-dimensional loading
Combinatorial optimization
Heuristics

A B S T R A C T

In this paper, we present heuristic algorithms for a three-dimensional loading capacitated vehicle routing problem arising in a real-world situation. In this problem, customers make requests of goods, which are packed in a sortment of boxes. The objective is to find minimum cost delivery routes for a set of identical vehicles that, departing from a depot, visit all customers only once and return to the depot. Apart of the usual 3D container loading constraints which ensure that the boxes are packed completely inside the vehicles and that the boxes do not overlap each other in each vehicle, the problem also takes into account constraints related to the vertical stability of the cargo and multi-drop situations. The algorithms are based on the combination of classical heuristics from both vehicle routing and container loading literatures, as well as two metaheuristic strategies, and their use in more elaborate procedures. Although these approaches cannot assure optimal solutions for the respective problems, they are relatively simple, fast enough to solve real instances, flexible enough to include other practical considerations, and normally assure relatively good solutions in acceptable computational times in practice. The approaches are also sufficiently generic to be embedded with algorithms other than those considered in this study, as well as they can be easily adapted to consider other practical constraints, such as the load bearing strength of the boxes, time windows and pickups and deliveries. Computational tests were performed with these methods considering instances based on the vehicle routing literature and actual customers' orders, as well as instances based on a real-world situation of a Brazilian carrier. The results show that the heuristics are able to produce relatively good solutions for real instances with hundreds of customers and thousands of boxes.

1. Introduction

The *three-dimensional loading capacitated vehicle routing problem* (3L-CVRP) is the result of the combination of the capacitated vehicle routing problem (CVRP) with the container loading problem (CLP). In this combined problem, generically, the challenge is to optimize simultaneously the planning of the vehicles' routes and the cargo arrangement inside the vehicle, while addressing a series of practical considerations inherited from both vehicle routing (Golden, Raghavan, & Wasil, 2008; Laporte, 2009; Toth & Vigo, 2002) and container loading (Bischoff & Ratcliff, 1995; Bortfeldt & Wäscher, 2013; Wäscher, Haussner, & Schumann, 2007) literatures.

This effort arises from the attempt to avoid expressing the demands of the customers simply as their weights or volumes. In other words, if the demand constraints are seen from a one-dimensional point of view, it is assumed that each demand fills a certain section of the vehicle or that the cargo shapes up smoothly according to the vehicle shape. However, when the demands are put in terms of rigid discrete items, such as boxes, their geometry may lead to losses of space or even to infeasible solutions if the vehicle has not enough capacity. If other practical constraints are also considered, the coupling of the routing and loading structures becomes even more complex.

The 3L-CVRP considers a fleet of identical vehicles that must run minimum cost routes to deliver boxes to a set of customers, departing from and returning to a depot. Besides the non-overlap of the three-dimensional boxes, the constraints that have been usually considered are the vertical stability of the cargo and the multi-drop situations (also known as LIFO constraints), although other constraints may also appear, such as the load bearing

☆ This manuscript was processed by Area Editor Mitsuo Gen.

* Corresponding author.
E-mail addresses: leonardo.junqueira@uninove.br (L. Junqueira), reinaldomorabito@gmail.com (R. Morabito).

http://dx.doi.org/10.1016/j.cie.2015.06.005
0360-8352/

strength of the boxes, time windows and pickups and deliveries. The approaches used to solve the problem have been mainly heuristic (Bortfeldt, 2012; Bortfeldt & Homberger, 2013; Ceschia, Schaerf, & Stützle 2013; Fuellerer, Doerner, Hartl, & Iori, 2010; Gendreau, Iori, Laporte, & Martello, 2006; Lacomme, Toussaint, & Duhamel, 2013; Moura & Oliveira, 2009; Ruan, Zhang, Miao, & Shen, 2013; Tao & Wang, 2013; Tarantilis, Zachariadis, & Kiranoudis, 2009; Zachariadis, Tarantilis, & Kiranoudis, 2012, 2013; Zhu, Qin, Lim, & Wang, 2012). The integrated problem is usually hierarchically decomposed, where the vehicle routing is solved at the higher level and the container loading at the lower level. Recent reviews on the emerging literature on these problems are presented in Iori and Martello (2010, 2013). Table 1 presents a compilation of the published papers that addressed some integration of vehicle routing and three-dimensional loading problems, focusing on the main features considered, which are abbreviated as follows:

C1 – Orientation constraints	C6 – Weight limit constraints
C2 – Cargo stability constraints	C7 – Time windows constraints
C3 – Multi-drop constraints	C8 – Time-constrained routes
C4 – Load bearing/fragility constrains	C9 – Pickup and delivery points
C5 – Boxes in pallets first, pallets in vehicles second	C10 – Split deliveries

In this paper, we present reasonably simple and effective heuristic algorithms for an integrated vehicle routing and three-dimensional loading problem arising in the context of a carrier. Apart of the geometrical constraints, that ensure that the boxes are packed completely inside the vehicles and that the boxes do not overlap each other in each vehicle, the vertical stability of the cargo (C2) and the multi-drop situations (C3) are also taken into account. Cargo stability (e.g., Fanslau & Bortfeldt, 2010; Gonçalves & Resende, 2012; Junqueira, Morabito, & Yamashita, 2012a; Zheng, Chien, & Gen, 2014; Zhu & Lim, 2012) refers to the support of the bottom faces of the boxes, in the case of vertical stability (i.e., the boxes must have their bottom faces supported by the top faces of other boxes or by the floor of the container), and to the support of the lateral faces of the boxes, in the case of horizontal stability. Multi-drop situations (e.g., Ceschia & Schaerf, 2013; Christensen & Rousøe, 2009; Junqueira, Morabito, & Yamashita, 2012b; Lai, Xue, & Xu, 1998) refer to cases where boxes that are delivered to the same customer (destination) must be placed close to each other in the vehicle, and the loading pattern must take into account the delivery route of the vehicle and the sequence in which the boxes are unloaded. The practical importance of incorporating these constraints to the problem is to avoid loading patterns where boxes are floating inside the vehicles or where an unnecessary additional handling is incurred when each drop-off point of the route is reached, which will become more evident by the real-world situation described in Section 2. It is also assumed that the boxes and the container/vehicle are of rectangular shape.

To our knowledge, the specialized literature has been concentrated in presenting solution methods that are able to deal with 3L-CVRP instances (Gendreau et al., 2006) with dozens of customers and dozens of boxes of relatively *many types* (i.e., strongly heterogeneous instances), which configures the integration of the CVRP with the 3D-R-SBSBPP (*three-dimensional rectangular single bin size bin packing problem*) or with the 3D-R-SKP (*three-dimensional rectangular single knapsack problem*), according to the typology presented in Wäscher et al. (2007), and which is a common scenario in courier services or when retailers need to perform deliveries of purchases in small quantities to final customers. In this paper, motivated by visits performed on a Brazilian carrier, we present solution methods that are able to deal with 3L-CVRP instances with hundreds of customers and thousands of boxes of relatively *few types* (i.e., weakly heterogeneous instances), which configures the integration of the CVRP with the 3D-R-SSSCSP (*three-dimensional rectangular single stock size cutting stock problem*) or with the 3D-R-SLOPP (*three-dimensional rectangular single large object placement problem*), according to the typology presented in Wäscher et al. (2007), and which is a common scenario in carrier services or when companies need to perform deliveries of batches of goods to warehouses or to retailers, or from warehouses to retailers. The sortment of boxes plays an important role in the design of packing heuristics, whereas packing algorithms designed for dealing with strongly heterogeneous sortments of boxes may not be suitable for dealing with weakly heterogeneous sortments of boxes, and vice versa. The proposed solution approaches take into account this gap, besides being relatively simple, fast enough to solve real instances, flexible enough to include other practical considerations, and normally assuring relatively good solutions in acceptable computational times in practice, still being sufficiently generic to be embedded with algorithms other than those considered in this study.

This work is organized as follows. In Section 2, we describe the 3L-CVRP arising in the context of a carrier. In Section 3, we review classical heuristics from both vehicle routing and container loading literatures, as well as two metaheuristic algorithms. In Section 4, we build on the heuristics presented in Section 3, extending and adapting them to address the 3L-CVRP, and we incorporate them in more elaborate procedures. In Section 5, we analyze the results

Table 1
Overview of constraints addressed in publications on integrated vehicle routing and three-dimensional loading problems.

References in alphabetical order	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
Bortfeldt (2012)	x	x	x	x		x				
Bortfeldt and Homberger (2013)	x	x	x				x			
Ceschia et al. (2013)	x	x	x	x		x				x
Fuellerer et al. (2010)	x	x	x	x		x				
Gendreau et al. (2006)	x	x	x	x		x				
Junqueira, Oliveira, Carravilla, and Morabito (2013)	x	x	x	x						
Lacomme et al. (2013)	x					x				
Moura and Oliveira (2009)	x	x	x				x			
Ruan et al. (2013)	x	x	x	x		x				
Tao and Wang (2013)	x	x	x	x		x				
Tarantilis et al. (2009)	x	x	x	x		x				
Zachariadis et al. (2012)	x	x			x		x	x		
Zachariadis et al. (2013)	x	x			x		x		x	
Zhu et al. (2012)	x	x	x	x		x				

Fig. 1. Overview of the arrival, dispatch and transfer courtyard in the Brazilian carrier.

of some computational tests with the proposed methods. Finally, in Section 6, we present concluding remarks and some perspectives for future research.

2. The problem in a carrier

Fig. 1 shows a picture of the company's cargo terminal in Guarulhos, located in the metropolitan area of São Paulo. In this cargo terminal, pickup and delivery services throughout this metropolitan area are performed, besides transfer services with other cargo terminals owned by the company. In this picture, the arrival of vehicles (after pickups) occurs in the docks on the left side, while the dispatch of vehicles (before deliveries) occurs in the docks on the right side. For these services smaller trucks are used. Fig. 2 on the left shows a typically used truck. There are also docks to the back of this picture that are used for the transfer services, in which larger vehicles are used. The courtyard in the middle is used for screening the cargo based on the origin–destination relation of each request. Fig. 2 on the right shows a conveyor device that is used to scan and assess the dimensions and weight of each box. The focus of this study remained on the short-haul routing, especially in what concerns the delivery operations. The cross-docking operations, as well as the transfer operations between terminals, were not addressed in this study and constitute an interesting topic for future research.

During the visits performed to the company's cargo terminal it was possible to note the high volume of goods transported all days. The turnover of these goods is also high as they rarely stay in the cargo terminal more than 24 h. The goods normally arrive packed in rectangular boxes and the delivery operations throughout the coverage region normally do not involve boxes of many different shapes (i.e., a weakly heterogeneous sortment of boxes). The boxes picked up from the coverage region or received from other terminals are screened during the whole day and they are delivered to the coverage region or sent to other terminals in the next day.

The definition of the routes that the vehicles shall perform in the next day is done during the night. The company acquired a few years ago the software Roadnet®, a routing package currently developed and marketed by Roadnet Technologies, which makes use of different solution methods for Vehicle Routing Problems to help in decision making in real situations. The routes are then provided to "cargo assembly teams" during the whole dawn. These teams are made up of workers who arrange the cargo itself and of workers who verify the labels and bar codes on the boxes, which are also responsible for issuing the invoices to the drivers. Once the vehicle is loaded, the respective driver can then perform the assigned route.

As soon as a route is provided, the respective cargo assembly team normally tries to organize the arrangement of boxes according to the invoices, i.e., the workers empirically aim at placing the boxes of customers that are going to be visited later in the route closer to the vehicle's cabin, while placing the boxes of customers that are going to be visited earlier in the route closer to the vehicle's door. Other concerns include the proper support from below for all boxes. Despite the cargo assembly teams have large experience in performing this activity, it is not uncommon to occur that they end up not loading completely the boxes of one or more customers inside one or more vehicles. In this case, what the company normally does is to assign another smaller vehicle to perform the delivery of boxes that were left out, but this situation is undesirable. In other words, the solutions achieved by the company, despite being of good quality from the point of view of the vehicle routing, provide no guarantees that the resultant loadings are feasible in practice, as could be observed in some cases.

3. Classical routing and loading heuristics

In a broad manner, the procedures presented in Section 4 can be decomposed in a *routing module* and a *loading module*.

3.1. The routing module

The routing module basically consists of the Clarke and Wright and Gillett and Miller algorithms, for the generation of the routes, and of the 2-Opt and 1-Point local search moves for the improvement of the solution found. The choice for these algorithms is due to the fact that they are relatively simple to be implemented in practice and flexible enough to include other practical considerations, besides being present in many widely marketed routing packages. Two metaheuristic strategies, the Simulated Annealing and the Record-to-Record Travel algorithms, are also employed in order to avoid getting stuck in local optima.

3.1.1. Constructive vehicle routing heuristics

One of the heuristics that composes the routing module is the Clarke and Wright algorithm (Clarke & Wright, 1964), also commonly known as the *Savings* algorithm. This procedure consists in a constructive heuristic that is based on the "savings" concept, which basically expresses the economies of cost that may be obtained when two separate routes are joined in one single route. The other heuristic that composes this module is the Gillett and Miller algorithm (Gillett & Miller, 1974), also commonly known as the *Sweep* algorithm. This procedure consists in a constructive heuristic that is based on the "cluster first, route second" concept, which basically builds sets of customers in a first step, and then builds a route for each set in a second step.

3.1.2. Local search heuristics

In order to improve the quality of the initial solutions provided by the constructive heuristics of Section 3.1.1, two local search operators that consider neighborhood structures for the Vehicle Routing Problem singly were also employed. These are the 1-Point and the 2-Opt moves, and both operators perform either intra-route or inter-routes moves. In a 1-Point move it is tried to relocate a single node in an existing route to a new position on the same route, in the case of an intra-route move, or to a new position on another route, in the case of an inter-route move.

Fig. 3 shows examples of intra-route and inter-routes 1-Point moves, respectively.

In a 2-Opt move it is tried to replace two existing edges from the same route with two new edges, in the case of an intra-route move, or to replace two existing edges from different routes with two new edges, in the case of an inter-route move. Fig. 4 shows examples of intra-route and inter-routes 2-Opt moves, respectively.

3.1.3. Metaheuristic procedures

In this study we also employed two metaheuristic procedures in order to try to obtain better results for the CVRP. These are the Simulated Annealing and the Record-to-Record Travel algorithms. The Simulated Annealing algorithm (Kirkpatrick, Gelatt, & Vecchi, 1983) has its inspiration in the annealing metallurgical process, in which a material is first heated and then slowly cooled under controlled conditions, in order to increase the crystals size and to reduce their defects. This process helps to improve the resistance and durability of the material. The heating increases the energy of the atoms allowing them to move with higher freedom, while a slow cooling program allows a low energy state to be achieved. As an analogy, the algorithm aims at finding the minimum cost state in the search space. To this end, the acceptance strategy of new solutions uses a probabilistic function that turns itself stricter for the solutions costs as the running time progresses. Let T, L, I, η, N be respectively the initial temperature, the loop size, the number of iterations per loop, the cooling ratio and the size of the neighbor list. The pseudocode of the employed Simulated Annealing algorithm is shown next (Algorithm 1):

Algorithm 1: Simulated annealing

Inputs: A initial feasible solution for the CVRP.
Outputs: An improved feasible solution for the CVRP.
1. Set $\{T; L; I; \eta; N\} = \{-; -; -; -; -\}$.
2. Set U = {1-Point, 2-Opt} the set of local search operators.
3. Set S equal to the current total traveled distance.
4. **for** $j = 1, \ldots, L$ **do**
5. $T = \eta \cdot T$.
6. **for** $k = 1, \ldots, I$ **do**
7. **for** all local search operators $u \in U$ **do**
8. **for** all nodes j in the solution **do**
9. Apply local search operator u on node j.
10. **if** the generated solution is better than the incumbent solution **then**
11. Accept the generated solution as the new incumbent solution.
12. **else if**
13. Accept the generated solution as the new incumbent solution if $e^{-(\text{candidate solution distance} - \text{current solution distance})/T}$ is larger than a random number drawn in the interval $(0; 1)$.
14. **end if**
15. **end for**
16. **end for**
17. **end for**
18. **end for**

The Record-to-Record Travel algorithm (Dueck, 1993) can be seen as a deterministic version of the Simulated Annealing algorithm. It alternates between a diversification phase and an improvement phase. In the diversification phase the objective is to accept some moves that may worsen (but not that much) the solution found so far, and to try to explore new regions of the search space. This phase is followed by an improvement phase, in which only moves that improve the solution are allowed, as the algorithm seeks for a local minimum in the search space. The algorithm ends after being unable to escape from a local minimum after a number of attempts. Let D, K, P, δ, N be respectively the loop size in the diversification phase, the number of local minima that shall be seeked before perturbing the solution or quitting the algorithm, the number of times the solution is perturbed once the search is stuck in a local minimum, the objective function deterioration allowed in the diversification phase and the size of the neighbor list. The pseudocode of the employed Record-to-Record Travel algorithm is shown next (Algorithm 2):

Algorithm 2: Record-to-Record Travel

Inputs: A initial feasible solution for the CVRP.
Outputs: An improved feasible solution for the CVRP.
1. Set $\{D; K; P; \delta; N\} = \{-; -; -; -; -\}$.
2. Set U = {1-Point, 2-Opt} the set of local search operators.
3. Set the record R equal to the current total traveled distance, set the tolerance $T = (1 + \delta) \cdot R$, and set $k = p = 0$.
4. **while** $p < P$ **do**
5. **for** $i = 1, \ldots, D$ **do**
6. **for** all local search operators $u \in U$ **do**
7. **for** all nodes j in the solution **do**
8. Apply the local search operator u on node j accepting improvement moves, or, in case none exists, accept the best deteriorating move if the resultant objective function is less than T.
9. **end for**
10. **end for**
11. **end for**
12. **while** improvement moves can be found **do**
13. **for** all local search operators **do**
14. **for** all nodes j in the solution **do**
15. Apply the local search operator u on node j accepting only improvement moves.
16. **end for**
17. **end for**
18. **end while**
19. **if** the current solution is a new record **then**
20. Update R and T and set $k = 0$.
21. **end if**
22. $k = k + 1$.
23. **if** $k == K$ **then**
24. Perturb the solution.
25. $p = p + 1$.
26. **end if**
27. **end while**

3.2. The loading module

The loading module basically consists of the George and Robinson algorithm, properly adapted to the context of the carrier problem addressed in this study, and of five variations of this algorithm also employed in order to verify the feasibility of the loading patterns. The choice for this algorithm is due to the fact that it is relatively simple to be implemented in practice and fast enough to be called many times per iteration, besides being more suitable for dealing with weakly heterogeneous sortments of boxes, which is the case of the scenario found in the carrier.

3.2.1. A constructive container loading heuristic

One of the heuristics that composes the loading module is the George and Robinson algorithm (George & Robinson, 1980). This

Fig. 2. Typically used truck and conveyor device for accessing the boxes' data.

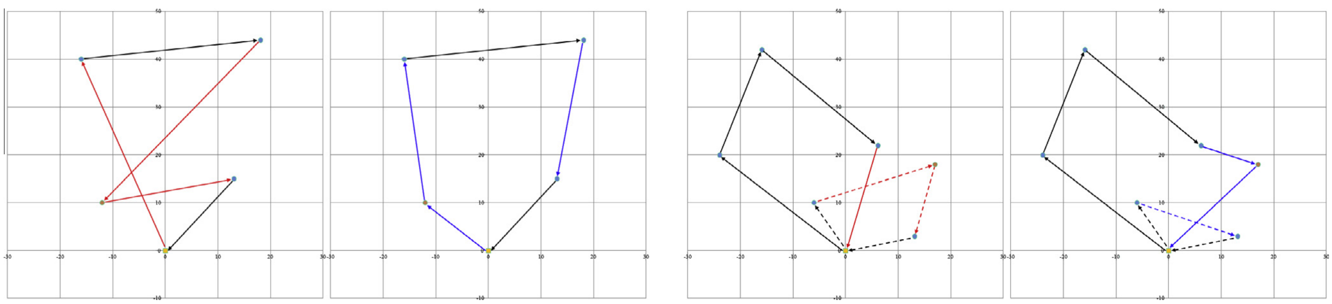

Fig. 3. Intra-route and inter-routes 1-Point moves.

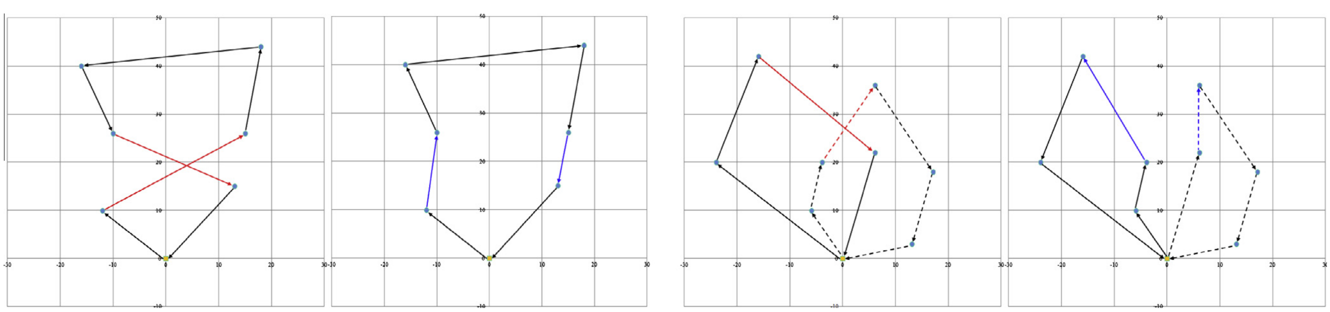

Fig. 4. Intra-route and inter-routes 2-Opt moves.

procedure consists in a constructive heuristic that is based on the building of *virtual walls* (vertical layers). This algorithm considers geometrical constraints and implicitly ensures vertical stability constraints. There is no restriction concerning the boxes orientation or the load bearing/fragility, although the algorithm can be easily adapted to address these situations. The algorithm fills the container/vehicle by building layers along its length L, from near the cabin to the door, trying to combine empty spaces between layers to improve the utilization of the available space. A layer is defined as a section of the container/vehicle's length in its complete width W and height H. The length of each layer is determined by the length of the box type that is chosen to start the filling of the layer. Fig. 5 shows three layers that are obtained with this wall building approach (different colors denote different box types).

The algorithm packs columns made of boxes of the same type. Each time a type is used it is given priority over the other, which induces boxes of the same type to be packed closer each other. The boxes types are classified in *not open*, if the type was not already used, *open*, if the type was already used, or *closed*, if there are no more boxes of the type. When there are *open* types, the one with the highest priority is selected to start a new layer. When there are no *open* types, it is selected the *not open* type with highest priority. The priority criteria are: (i) the box type with the largest of the smaller dimensions; (ii) the box type with the largest available quantity; (iii) the box type with the largest of the larger dimensions. It is used preferably criterion (i), and criteria (ii) and (iii) in cases of ties.

An important feature of this algorithm is that it introduces the idea of "flexible width", which is used when considering the use of not filled spaces of previous layers while building new layers. Available spaces of the current layer can be joined (i.e., "amalgamated") to not filled spaces of the previous layer. The flexible width determines the extent along the width in which this new space can be increased. This limitation allows successive attempts of making use of not filled spaces of previous layers and, as a consequence, it maintains the "contour" of the previous layer and avoids that empty spaces are not filled. The pseudocodes that compose this algorithm can be found in the original study of George and Robinson (1980) and are the base of the loading module.

3.2.2. Pool of container loading heuristics

The idea of the other heuristics that compose the loading module was firstly investigated by Cecilio and Morabito (2004). These heuristics consist of five variations of the original George

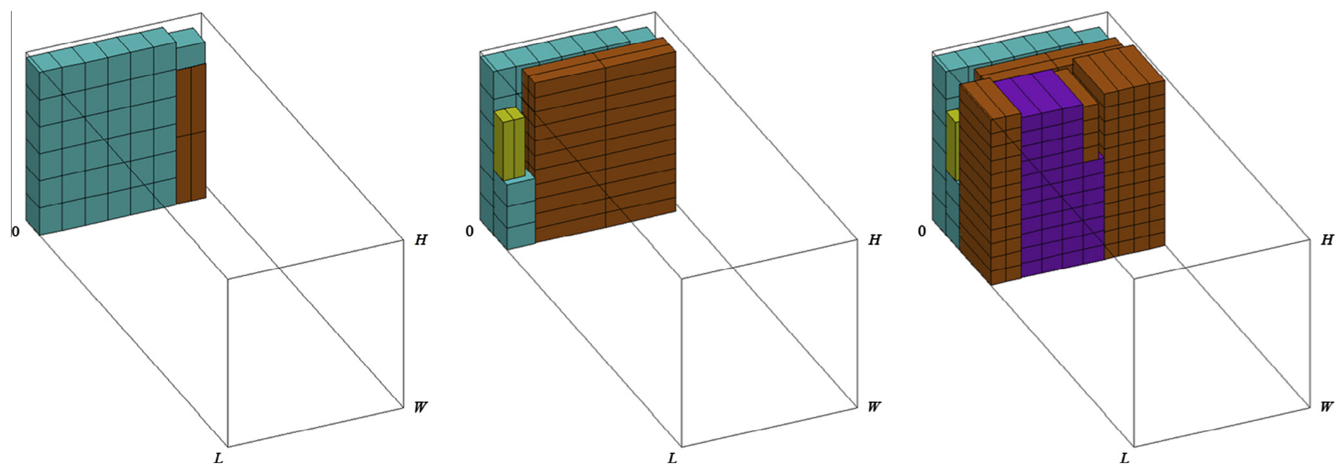

Fig. 5. Wall building with the original George and Robinson algorithm.

and Robinson algorithm. The first variation, called "Refinement", consists in a simple change in the original procedure that chooses the box type to fill the spaces other than a new layer. When verifying if there is a not closed box type that fills more than one complete column in the selected space, instead of simply choosing the box type that best fills the base area of the space, it is verified if a combination of identical boxes along the length of the space can be made. From this combination, it is chosen the box type that results in the best utilization of the base area of the space.

The second and fourth variations basically consist in changing the sequence of priority criteria that are used when a box type is selected to start filling a new layer. To this end, besides the three priority criteria of the original George and Robinson algorithm (see Section 3.2.1), two additional criteria are included: (iv) the box type with the largest volume; (v) the box type with the largest ratio (largest dimension)/(smallest dimension). Both variations use a permutation of three out of the five priority criteria (i.e., $5!/2!$ combinations), which results in sixty different sequences of priority criteria.

As mentioned, the second variation, called "Permutation", performs sixty times the George and Robinson algorithm to determine the best loading pattern for the whole container/vehicle, i.e., in each iteration a different permutation of three out of the five priority criteria is used, exhausting all the possible permutations. The best result in terms of the total volume packed is updated in each iteration. The third variation basically consists of the second variation combined with the Refinement procedure.

The fourth variation, called "Layer", performs the George and Robinson algorithm to determine the best loading pattern for each layer, i.e., in each iteration a different permutation of three out of the five priority criteria is used, exhausting all the possible permutations. The best result for each layer is determined by the ratio between the volume of boxes packed and the partial used volume of the container/vehicle, i.e., the result is as better as denser the layer is. The fifth variation basically consists of the fourth variation combined with the Refinement procedure.

3.2.3. Adaptation to the carrier problem

Despite implicitly ensuring vertical stability constraints, it is important to note here that the original George and Robinson algorithm and the five variations previoulsy described need to be adapted to address the multi-drop constraints that arise in the context of the carrier problem. The resultant pseudocodes remain very similar to the original ones, except for some small changes. In the case of the original George and Robinson algorithm, it is iteratively called to verify the feasibility of the packing pattern for each route. The algorithm starts with the set of boxes of the last customer to be visited in a given route. Once this set of boxes is packed, then it starts to pack the set of boxes of the penultimate customer to be visited in that route, and then successively until the set of boxes of the first customer to be visited in the route. Instead of always starting a new layer for each customer, the modified algorithm verifies if the set of boxes of a given customer fits in the empty spaces that were left over in the previous layer. At the end of the packing of any of the customers' boxes, if the number of packed boxes is lower than the number of required boxes, then the packing is infeasible and it is not necessary to consider the sets of boxes of the remaining customers. Fig. 6 shows the partial packing patterns that are obtained with the modified George and Robinson algorithm after loading the sets of boxes of three customers (different colors denote different customers).

In the case of the Permutation algorithm (i.e., the second variation), the adaptation is similar to the case of the original George and Robinson algorithm, except that the former is called sixty times to verify the feasibility of the packing pattern for each route. In a given iteration, at the end of the packing of any of the customers' boxes, if the number of packed boxes is lower than the number of required boxes, then the packing is infeasible and it is not necessary to consider in that iteration the sets of boxes of the remaining customers. Besides that, at the end of any of the sixty iterations, if the number of packed boxes is equal to the number of required boxes, then the packing is feasible and it is not necessary to consider the remaining iterations.

In the case of the Layer algorithm (i.e., the fourth variation), the adaptation is a bit more complicated since each layer may contain boxes of more than one customer. For example, after sixty iterations, if the best (i.e., the denser) first layer contains all boxes of the last customer to be visited in a given route, plus some of the boxes of the penultimate customer to be visited in that route, then the determination of the best second layer necessarily needs to start with the remaining boxes of the penultimate customer that were left over. That is, the sequence of customers and the number of their remaining boxes need to be updated throughout the filling of the container/vehicle. At the end of the procedure, if the number of packed boxes is lower than the number of required boxes, then the packing is infeasible.

The feasibility of the packing pattern in each container/vehicle is then sequentially verified by the modified original George and Robinson algorithm and its five variations previously described, that is, the George and Robinson algorithm with the Refinement procedure, the Permutation algorithm, the Permutation algorithm with the Refinement procedure, the Layer algorithm and the

Layer algorithm with the Refinement procedure. If the feasibility of the packing pattern is attested by any of these algorithms, then the remaining algorithms need not be called.

4. Solution strategies

The two main solution strategies presented in this section involve the calling of the routing and loading modules presented in Section 3 either sequentially or simultaneously. A third solution strategy is also presented that aims to combine features of the first two strategies. Note that the approaches that follow are sufficiently generic to be embedded with algorithms other than those considered in Section 3.

4.1. Strategy 1 – loading after routing

The first solution strategy evaluates the generation of the routes and the feasibility of the packing patterns singly and sequentially. The algorithm is based on the definition of a *stowage loss* parameter as an input data of the problem. Stowage loss $\theta \in [0,1]$ is defined as a fraction of the empty space not filled in a container, and it is an output of the Container Loading Problem. On the other hand, solutions for the Vehicle Routing Problem that contain vehicles with high capacity utilization are possibly infeasible when the respective Container Loading Problem is solved. As mentioned before, it happens because the demands are put in terms of rigid discrete items (i.e., boxes), and it is unlikely that the cargo shapes up smoothly according to the vehicles shape, besides having to consider additional constraints, such as vertical stability and multi-dropping.

This strategy consists in anticipating possibly infeasible solutions and determining *a priori* the stowage loss value. For example if the vehicles have a nominal capacity of $C = 100$, and it is supposed that the stowage loss is 10% on average, then the routing algorithms could generate solutions considering all vehicles with capacity $C' = 90$, instead of $C = 100$, as an attempt that the resultant packing patterns verified by the packing algorithms are all feasible considering the nominal capacity $C = 100$. If at least one of the resultant packing patterns verified by the packing algorithms is infeasible, then the stowage loss value is increased to, e.g., 15%, and the routing algorithms would generate solutions considering all vehicles with capacity $C'' = 85$. On the other hand, if all the resultant packing patterns verified by the packing algorithms are feasible, then the stowage loss value is decreased to, e.g., 5%, and the routing algorithms would generate solutions considering all vehicles with capacity $C'' = 95$. This procedure of increasing or decreasing the stowage loss value would be performed as an attempt to obtain increasingly better feasible solutions for the integrated problem, either trying to obtain feasible (or of better quality) solutions for the Container Loading Problem, or trying to obtain feasible (or of better quality) solutions for the Vehicle Routing Problem. The procedure ends up after not being more possible to locally improve a resultant solution for the Vehicle Routing Problem without incurring in an infeasible solution for the Container Loading Problem. Note that the continuous adaptation of the stowage loss value tries to balance the trade-off between the two problems, i.e., on one hand it tries to obtain good routes from the point of view of the total traveled distance, while possibly incurring in infeasible packing patterns, and on the other hand it tries to obtain feasible packing patterns, while possibly incurring in bad routes from the point of view of the total traveled distance. Let *CUL* be the capacity upper limit, *CLL* be the capacity lower limit and ε be a sufficiently small number (e.g., 10^{-5}). The pseudocode of the Stowage Loss algorithm is shown next (Algorithm 3):

Algorithm 3: Stowage loss

Inputs: A set of customers, with known positions and demands, and that shall be visited by a single vehicle with known capacity; A set of boxes to be packed inside the vehicles, with known types, dimensions e quantities (per customer); A stowage loss θ.
Outputs: A feasible solution for the 3L-CVRP.
1. Set *CUL* equal to the nominal capacity *C* and set *CLL* equal to 0.
2. Generate a set of routes by using the *Clarke and Wright* algorithm OR the *Gillett and Miller* algorithm.
3. Apply the *Local Search* operators OR the *Simulated Annealing* algorithm OR the *Record-to-Record Travel* algorithm on the solution found.
4. **for** each generated route **do**
5.   Generate the packing pattern by using the *George and Robinson* algorithm and the five variations discussed in Section 3.
6. **end for**
7. **if** at least one of the resultant packing patterns is not feasible **then**
8. $C = (1-\theta)\cdot C$.
9.   **while** $CUL - CLL > \varepsilon$ **do**
10.   Generate a set of routes by using the *Clarke and Wright* algorithm OR the *Gillett and Miller* algorithm.
11.   Apply the *Local Search* operators OR the *Simulated Annealing* algorithm OR the *Record-to-Record Travel* algorithm on the solution found.
12.   **for** each generated route **do**
13.   Generate the packing pattern by using the *George and Robinson* algorithm and the five variations discussed in Section 3.
14.   **end for**
15.   **if** at least one of the resultant packing patterns is not feasible **then**
16. $CUL = C$.
17. $C = (CUL + CLL)/2$.
18.   **else if**
19. $CLL = C$.
20. $C = (CUL + CLL)/2$.
21.   **end if**
22.   **end while**
23. **end if**

Fig. 7 shows an example involving Strategy 1. At the beginning the routing algorithms are employed with a stowage loss value of $\theta = 0$ for all vehicles, which is equivalent to considering 100% of the vehicles nominal capacity (see point a). For each of the resultant routes, at least one of the respective packing patterns verified by the packing algorithms is infeasible. Therewith the routing algorithms are employed again with a stowage loss value of $\theta = 0.50$ for all vehicles, which is equivalent to considering 50% of the vehicles nominal capacity (see point b). The red plane denotes an abstraction for the capacity limit that the routing algorithms consider at this point. For each of the resultant routes, all the respective packing patterns verified by the packing algorithms are feasible. Therewith the routing algorithms are employed again with a stowage loss value of $(a+b)/2$ for all vehicles, which is equivalent to considering 75% of the vehicles nominal capacity (see point c). For each of the

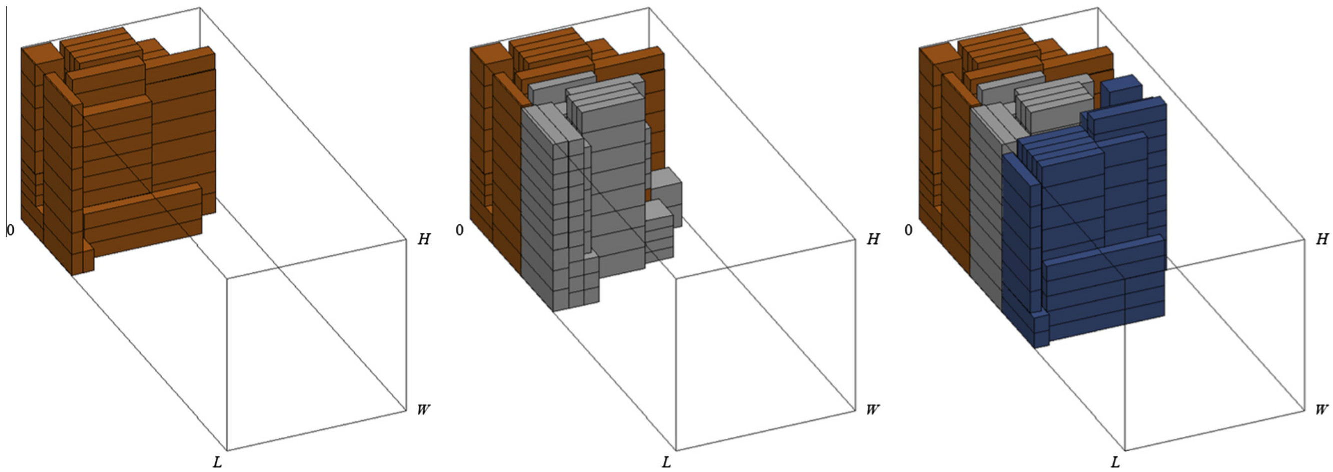

Fig. 6. Packing patterns with the modified George and Robinson algorithm.

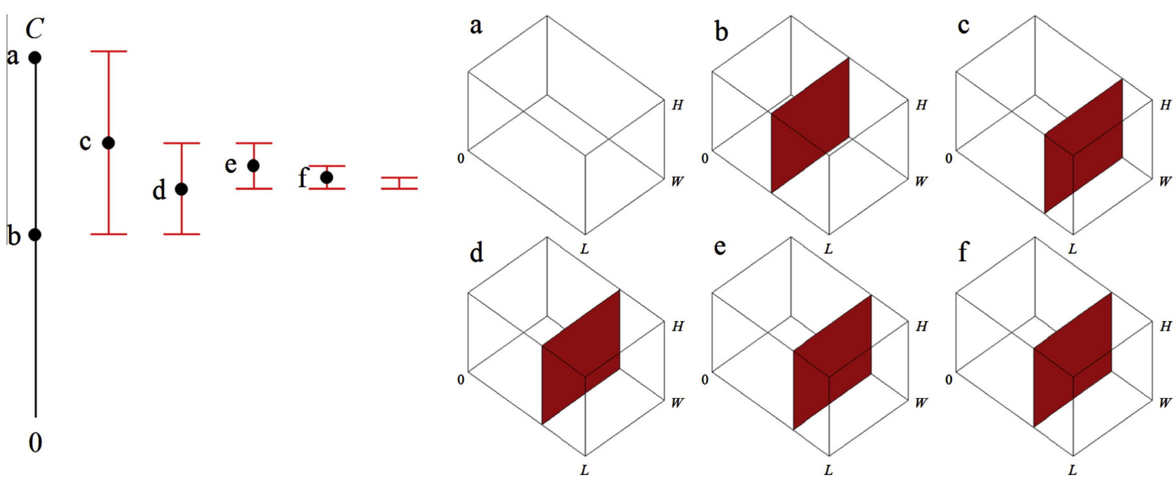

Fig. 7. Strategy 1 – loading after routing.

resultant routes, at least one of the respective packing patterns verified by the packing algorithms is infeasible. Therewith the routing algorithms are employed again with a stowage loss value of $(b+c)/2$ for all vehicles, which is equivalent to considering 62.5% of the vehicles nominal capacity (see point d). For each of the resultant routes, all the respective packing patterns verified by the packing algorithms are feasible. Therewith the routing algorithms are employed again with a stowage loss value of $(c+d)/2$ for all vehicles, which is equivalent to considering 68.25% of the vehicles nominal capacity (see point e). For each of the resultant routes, at least one of the respective packing patterns verified by the packing algorithms is infeasible. Therewith the routing algorithms are employed again with a stowage loss value of $(d+e)/2$ for all vehicles, which is equivalent to considering 65.625% of the vehicles nominal capacity (see point f). For each of the resultant routes, at least one of the respective packing patterns verified by the packing algorithms is infeasible. The algorithm continues until $CUL - CLL \leqslant \varepsilon$, which constitutes the stopping criterion for this strategy.

4.2. Strategy 2 – loading while routing

The second solution strategy evaluates the generation of the routes and the feasibility of the packing patterns simultaneously. The algorithm consists in verifying in every attempt of joining routes (in the case of the Clarke and Wright algorithm) or of assigning customers to vehicles (in the case of the Gillett and Miller algorithm), if the resultant packing pattern is feasible or not. However, at the beginning of the routes construction, it is unlikely that the resultant packing patterns are infeasible, since the vehicles are less filled and normally there is enough space for the proper placement of boxes. Therefore to avoid calling the packing algorithms (i.e., the original George and Robinson algorithm and the five variations discussed in Section 3) an unnecessary number of times, what would increase the computational times, we define a *tolerance* parameter $\tau \in [0, 1]$ as the fraction of the vehicles filled volume beyond which the packing algorithms are called to verify the feasibility of the packing patterns.

That is, when joining two routes, or when assigning a customer to a vehicle, if the filled volume is larger than $\tau \cdot C$, then the packing algorithms are called. For example, for a given tolerance of $\tau = 0.60$ (i.e., 60%), each route shall be verified by the packing algorithms after its filled volume surpasses 60% of the vehicle nominal capacity C. Note that, on one hand, if $\tau = 0$, the packing algorithms are always called, what would lead to excessive computational times, and on the other hand, if $\tau = 1$, the packing algorithms are never called, what would possibly lead to infeasible packing patterns.

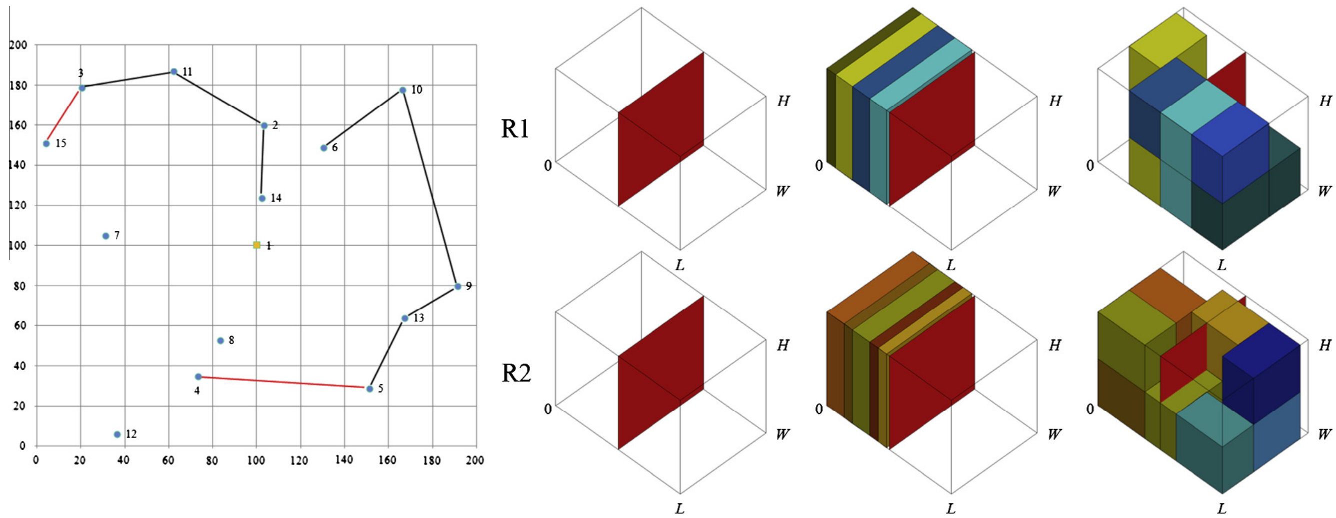

Fig. 8. Strategy 2 – loading while routing.

The pseudocodes of the Tolerance algorithm embedded with the Clarke and Wright algorithm (Algorithm 4) or with the Gillett and Miller algorithm (Algorithm 5) are shown next:

Algorithm 4: Tolerance (for the Clarke and Wright algorithm)

Inputs: A set of customers, with known positions and demands, and that shall be visited by a single vehicle with known capacity; A set of boxes to be packed inside the vehicles, with known types, dimensions e quantities (per customer); A tolerance τ.
Outputs: A feasible solution for the 3L-CVRP.
1. Generate a route for each customer.
2. **for** all pairs of customers **do**
3. Calculate the values of the savings and put them in a list.
4. **end for**
5. Put the list in non-increasing order of the savings values.
6. **while** it is possible to join routes **do**
7. Start with the pair of customers with the largest savings value.
8. **if** the packed volume of the resultant route is larger than $\tau \cdot C$ **then**
9. Generate the packing pattern by using the *George and Robinson* algorithm and the five variations discussed in Section 3.
10. **if** the resultant packing pattern is feasible **then**
11. Join the routes.
12. **end if**
13. **else if**
14. Join the routes.
15. **end if**
16. Remove from the list the pair of customers and the respective saving.
17. **end while**
18. Apply the *Local Search* operators OR the *Simulated Annealing* algorithm OR the *Record-to-Record Travel* algorithm on the solution found, while keeping the feasibility of all packing patterns generated during execution.

Algorithm 5: Tolerance (for the Gillett and Miller algorithm)

Inputs: A set of customers, with known positions and demands, and that shall be visited by a single vehicle with known capacity; A set of boxes to be packed inside the vehicles, with known types, dimensions e quantities (per customer); A tolerance τ.
Outputs: A feasible solution for the 3L-CVRP.
1. **while** there are non-routed customers **do**
2. Choose a vehicle still not used.
3. **while** the vehicle capacity is not exceeded **do**
4. Start with the customer still not routed with the smallest angle.
5. **if** the packed volume of the resultant route is larger than $\tau \cdot C$ **then**
6. Generate the packing pattern by using the *George and Robinson* algorithm and the five variations discussed in Section 3.
7. **if** the resultant packing pattern is feasible **then**
8. Assign the customer to the vehicle.
9. **end if**
10. **else if**
11. Assign the customer to the vehicle.
12. **end if**
13. **end while**
14. **end while**
15. Apply the *Local Search* operators OR the *Simulated Annealing* algorithm OR the *Record-to-Record Travel* algorithm on the solution found, while keeping the feasibility of all packing patterns generated during execution.

Fig. 8 shows an example involving Strategy 2. At the beginning, the routing algorithms are employed without considering the loading constraints. Therewith are generated two routes R1 and R2. The red plane denotes an abstraction for the loading tolerance limit which, once surpassed, "activates" the calling of the packing algorithms. In this case the tolerance value is $\tau = 0.50$, which is equivalent to waiting each vehicle becomes 50% filled to start verifying if the respective packing pattern is feasible or not. This situation persists until customer "3" is included in route R1 and customer "5" is included in route R2. At this moment the routes are: R1 (1-3-11-2-14-1) and R2 (1-5-13-9-10-6-1) (see the respective partial packing patterns). For the sake of simplicity, the arcs connecting the customers and the depot are not shown. The inclusion of customer "15" in route R1 makes the 50% of tolerance limit of the nominal capacity to be surpassed, which activates the packing algorithms to verify if this inclusion results in a feasible packing pattern or not. The same occurs when customer "4" is included in route R2. In this case both resultant packing patterns are feasible, but from this point on for each route the packing algorithms are always called, once the tolerance value was surpassed. At this moment the routes are: R1 (1-15-3-11-2-14-1) and R2 (1-4-5-13-9-10-6-1). The algorithm continues according to each routing algorithm (i.e., the Clarke and Wright or the Gillett and Miller algorithm), i.e., the stopping criteria for this strategy follow the stopping criteria of the routing algorithms (unless they have an improvement phase after the construction phase). Note that if applying the local search operators, the tolerance rule remains, i.e., the feasibility of the packing patterns is always verified if the move results in a route whose vehicle has the tolerance value surpassed.

4.3. Strategy 3 – mixed strategies 1 and 2

The third solution strategy basically consists in using the solution generated by Strategy 1 as the initial solution for the metaheuristic procedures (i.e., the Simulated Annealing or the Record-to-Record Travel algorithms), but having the concern of maintaining the feasibility of the packing patterns when applying the local search operators, thus calling the packing algorithms whenever the filled volume inside each vehicle is larger than $\tau \cdot C$. The pseudocode of the hybrid Stowage Loss + Tolerance algorithm is shown next (Algorithm 6):

Algorithm 6: Stowage Loss + Tolerance

Inputs: A set of customers, with known positions and demands, and that shall be visited by a single vehicle with known capacity; A set of boxes to be packed inside the vehicles, with known types, dimensions e quantities (per customer); A stowage loss θ; A tolerance τ.
Outputs: An improved feasible solution for the 3L-CVRP.

1. Set *CUL* equal to the nominal capacity *C* and set *CLL* equal to 0.
2. Generate a set of routes by using the *Clarke and Wright* algorithm OR the *Gillett and Miller* algorithm.
3. Apply the *Simulated Annealing* algorithm OR the *Record-to-Record Travel* algorithm on the solution found.
4. **for** each generated route **do**
5. Generate the packing pattern by using the *George and Robinson* algorithm and the five variations discussed in Section 3.
6. **end for**
7. **if** at least one of the packing patterns is not feasible **then**
8. $C = (1-\theta) \cdot C$.
9. **while** $CUL - CLL > \varepsilon$ **do**
10. Generate a set of routes by using the *Clarke and Wright* algorithm OR the *Gillett and Miller* algorithm.
11. Apply the *Simulated Annealing* algorithm OR the *Record-to-Record Travel* algorithm on the solution found.
12. **for** each generated route **do**
13. Generate the packing pattern by using the *George and Robinson* algorithm and the five variations discussed in Section 3.
14. **end for**
15. **if** at least one of the packing patterns is not feasible **then**
16. $CUL = C$.
17. $C = (CUL + CLL)/2$.
18. **else if**
19. $CLL = C$.
20. $C = (CUL + CLL)/2$.
21. **end if**
22. **end while**
23. **end if**
24. Apply the *Simulated Annealing* algorithm OR the *Record-to-Record Travel* algorithm on the solution found, while keeping the feasibility of all packing patterns generated during execution.

Note that in the first applications of the metaheuristic procedure (i.e., the Simulated Annealing or the Record-to-Record Travel algorithm) (lines 3 and 11) there is no concern in keeping or even verifying the feasibility of the packing patterns in the solutions generated, while in its last application (line 24) the feasibility of the packing patterns is always verified and kept whenever the filled volume inside each vehicle is larger than $\tau \cdot C$.

5. Computational results

The algorithms presented in Sections 3 and 4 were written in C++, and part of the source code was provided by the VRPH library of heuristics (Groër, Golden, & Wasil, 2010), mainly the algorithms that constitute the routing module. All computational tests were performed in a PC Core i7 (2.10 GHz, 8.0 GB), running Windows. To evaluate their performances the algorithms were tested with instances based on the vehicle routing literature and actual customers' orders, as well as instances based on data collected at the Brazilian carrier (see Section 2).

When appropriate, the stowage loss value was set as $\theta = 0.50$ (Strategies 1 and 3), and the tolerance value was set as $\tau = 0.50$ (Strategies 2 and 3). These intermediate values for these parameters were set on a common sense basis, since, from the point of view of the arrangement of the cargo, a very small value for θ would possibly retard obtaining an initial feasible solution, and a very large value for τ would possibly not avoid the risk of obtaining infeasible solutions. The parameters for the Simulated Annealing algorithm were set as $\{T; L; I; \eta; N\} = \{2; 200; 2; 0.99; 10\}$, while the parameters for the Record-to-Record Travel algorithm were set as $\{D; K; P; \delta; N\} = \{30; 5; 1; 0.01; 40\}$, as they are tuned in the VRPH library of heuristics. The vehicles considered have dimensions (in millimeters) $(L, W, H) = (7320, 2480, 2630)$ and they are equivalent to a single-axis box truck with up to 6000 kg of weight capacity. The boxes types were obtained with the carrier as it uses a conveyor device that scans and assesses the dimensions and weight of each box (see Fig. 2). In the experiments that consider three-dimensional loading constraints, the vertical stability of the cargo and the multi-drop situations are always verified. In the case of the vertical stability of the cargo, 100% of the base area of each box must be supported (i.e., full support), which is always guaranteed by the George and Robinson algorithm and the five variations discussed in Section 3. In the case of the multi-drop situations, boxes belonging to a customer that is visited later in a route cannot block access to any box belonging to a customer that is visited earlier in the same route, which is always guaranteed by the adapted

Table 2
Instances based on Christofides et al. (1979).

Ins.	No. Cus.	No. Typ.	No. Box.
1	50	5	2366
2	75	5	4751
3	100	5	3536
4	150	5	5417
5	199	5	7731
6	120	5	3335
7	100	5	4401

versions of the aforementioned algorithms. In the tables that follow the abbreviations of the algorithms refer to:

CW: Clarke and Wright algorithm	LS: Local Search operators
GM: Gillett and Miller algorithm	SA: Simulated Annealing algorithm
GR: George and Robinson algorithm and its variations	RR: Record-to-Record Travel algorithm

Tables 3, 4, 13 and 14 present the results obtained for the algorithms, based either on the Clarke and Wright or the Gillett and Miller algorithm, when no three-dimensional loading constraint is considered (i.e., it is equivalent to solving the CVRP with volumetric capacity constraints only). Tables 5–10 and 15–20 present the results obtained for the algorithms, based either on the Clarke and Wright or the Gillett and Miller algorithm, when three-dimensional loading constraints are considered (i.e., the carrier problem considering vertical stability of the cargo and multi-drop situations).

In general, all these tables present for each instance (Ins.) and for each of the algorithms employed: the objective function value (OF), the number of routes/vehicles (No. Rou.), the gap or deviation (Gap or Dev., in%), the computational time (Time, in seconds), and the maximum density value (Max Den., in%) obtained (i.e., the most densely packed vehicle). These tables also present average values for the gap/deviation, the computational time and the maximum density value, for each set of instances and algorithm employed. That is, Tables 3, 5, 7, 9, 13, 15, 17 and 19 present results that are based on the Clarke and Wright algorithm, while Tables 4, 6, 8, 10, 14, 16, 18 and 20 present results that are based on the Gillett and Miller algorithm. Still in Tables 5–10 and 15–20, for each instance: (i) the results highlighted in italics denote the best results obtained in each strategy, based either on the Clarke and Wright or the Gillett and Miller algorithm; (ii) the results highlighted in bold denote the best general results obtained in each strategy; and (iii) the results marked with an asterisk (*) denote the best results among all strategies.

All these tables follow a same format, i.e., they show (from left to right) the results obtained with the constructive heuristic (either the Clarke and Wright or the Gillett and Miller algorithm), then the results obtained with the constructive heuristics and the local search operators, and finally the results obtained with the metaheuristic procedures (the Simulated Annealing and the Record-to-Record Travel algorithms). The idea is to evaluate the performance of the different strategies and algorithms.

5.1. Experiments on instances based on the literature

In a first set of experiments it was considered seven out of the fourteen instances proposed by Christofides, Mingozzi, and Toth (1979) for the CVRP. Seven of these instances (the ones considered in this study) only consider capacity constraints, while the other seven instances (that respectively share the same graphs of the preceding ones) further consider distance-constrained routes. The instances consider symmetric costs with the number of nodes n varying from 51 to 200. The coordinates of node 1 (i.e., the depot) lie always on position (0;0) of a Cartesian coordinate system, while the coordinates of the remaining nodes (i.e., the customers) lie scattered around the depot (except for instance 6, which has the customers more concentrated in one side of the depot). It was considered in all instances the same 5 box types with dimensions in millimeters, obtained from the carrier. The total number of boxes varies from 2366 to 7731. Table 2 presents, for each instance (Ins.), the number of customers (No. Cus.), the number of box types (No. Typ.) and the total number of boxes (No. Box.). In such a way to make use of the known lower bounds of the vehicle routing literature, for each of these instances we made a relation between the vehicles original capacity, as in Christofides et al. (1979), and the volume of the vehicles considered. Based on this relation it was possible to obtain new demand values (in terms of volume) for each customer. However, giving the difficulty of finding a set of boxes of the 5 considered box types whose sum is exactly equal to each new demand value, we proceeded to randomly draw the boxes to each customer, in such a way that the draw was stopped as soon as each new demand value was minimally surpassed. These demand values obtained from these drawn were the ones considered for this set of instances. Note therefore that the resultant instances are not exactly the same as the ones proposed by Christofides et al. (1979), since the demands considered here are slightly larger.

Tables 3 and 4 also present, for each instance, the objective function value (OF) and the number of routes/vehicles (No. Rou.) present in the best known solution from the literature, as reported by Cordeau, Gendreau, Laporte, Potvin, and Semet (2002). That is, without considering the arrangement of the cargo inside the vehicles and assuming therefore that the stowage loss value is equal to zero. It implies that the solutions obtained by the literature for these instances have no guarantees to be feasible from the point of view of the arrangement of the cargo (and they are probably infeasible). Some computational times for the literature methods to obtain these solutions are also reported by Cordeau et al. (2002), as well as some specifications of the hardware used. Note that the gaps (Gap, in%) in Tables 3 and 4 are obtained with relation to the best known solution from the literature, while the deviations (Dev., in%) in Tables 5–10 are obtained with relation to the best solution (among all algorithms proposed) obtained without considering three-dimensional loading constraints (see Tables 3 and 4, values highlighted in bold). Both values are computed similarly as follows:

$$\frac{(\text{best obtained} - \text{best literature})}{(\text{best literature})}100\%$$

$$\text{or } \frac{(\text{best obtained with 3D} - \text{best obtained without 3D})}{(\text{best obtained without 3D})}100\%$$

For this set of instances, in terms of solution quality, the algorithm that had the best performance was the Simulated Annealing algorithm building on the Clarke and Wright algorithm, having obtained an average gap of 2.78% in an average computational time of 4.46 s. Then appears the Record-to-Record Travel algorithm building on the Clarke and Wright algorithm, having obtained an average gap of 3.24% in an average computational time of 2.03 s. It is also interesting to note the high values of maximum density that were obtained for all instances, which is an evidence that these solutions are probably not feasible from the point of view of the arrangement of the cargo.

In the case of Strategy 1 (see Tables 5 and 6, highlighted in bold), the Record-to-Record Travel algorithm building on the

Table 3
Results obtained with the Clarke and Wright-based algorithms with no three-dimensional loading constraint for the instances based on Christofides et al. (1979).

Ins.	Best Sol.		CW					CW + LS					CW + SA					CW + RR				
	OF	No. Rou.	OF	No. Rou.	Gap	Time	Max Den.	OF	No. Rou.	Gap	Time	Max Den.	OF	No. Rou.	Gap	Time	Max Den.	OF	No. Rou.	Gap	Time	Max Den.
1	524.61	5	592.09	6	12.86	0.00	99.74	572.40	6	9.11	0.00	99.74	537.76	6	2.51	1.00	99.73	538.50	6	2.65	0.56	98.51
2	835.26	10	936.33	11	12.10	0.00	99.94	921.33	11	10.30	0.02	99.94	858.52	11	2.78	2.12	99.80	863.80	11	3.42	1.40	99.86
3	826.14	8	877.36	8	6.20	0.00	99.63	867.69	8	5.03	0.03	99.63	832.25	8	0.74	3.52	99.93	843.26	8	2.07	1.92	99.59
4	1028.42	12	1144.58	12	11.29	0.00	99.92	1120.78	12	8.98	0.05	99.91	**1042.48**	12	1.37	6.77	99.48	1056.37	12	2.72	2.95	99.62
5	1291.29	16	1407.40	17	8.99	0.03	99.64	1397.52	17	8.23	0.08	99.64	1341.92	17	3.92	10.86	99.88	1342.87	17	3.99	4.70	99.85
6	1042.11	7	1090.15	8	4.61	0.01	99.46	1071.18	8	2.79	0.05	99.46	1066.80	8	2.37	3.85	99.28	**1063.42**	8	2.04	1.31	99.46
7	819.56	10	908.24	10	10.82	0.00	96.52	903.74	10	10.27	0.02	96.52	866.86	10	5.77	3.07	96.62	867.10	10	5.80	1.36	96.62
Avg.	–	–	–	–	9.55	0.01	99.26	–	–	7.82	0.04	99.26	–	–	2.78	4.46	99.25	–	–	3.24	2.03	99.07

Table 4
Results obtained with the Gillett and Miller-based algorithms with no three-dimensional loading constraint for the instances based on Christofides et al. (1979).

Ins.	Best Sol.		GM					GM + LS					GM + SA					GM + RR				
	OF	No. Rou.	OF	No. Rou.	Gap	Time	Max Den.	OF	No. Rou.	Gap	Time	Max Den.	OF	No. Rou.	Gap	Time	Max Den.	OF	No. Rou.	Gap	Time	Max Den.
1	524.61	5	719.37	6	37.12	0.00	98.85	570.12	6	8.68	0.01	98.85	538.50	6	2.65	1.06	98.51	**533.40**	5	1.68	0.55	99.94
2	835.26	10	1098.87	11	31.56	0.00	98.92	899.29	11	7.67	0.00	99.59	**850.37**	11	1.81	2.06	99.41	853.02	10	2.13	0.94	99.97
3	826.14	8	1181.86	8	43.06	0.00	96.68	856.89	8	3.72	0.03	99.70	**830.08**	8	0.48	3.51	99.84	830.13	8	0.48	2.09	99.27
4	1028.42	12	1697.59	13	65.07	0.00	98.67	1119.31	13	8.84	0.06	99.98	1048.48	13	1.95	6.82	99.82	1051.67	13	2.26	2.64	99.48
5	1291.29	16	2208.40	17	71.02	0.01	99.76	1428.91	17	10.66	0.11	99.89	**1338.07**	17	3.62	11.08	99.77	1338.47	17	3.65	5.57	99.50
6	1042.11	7	3003.99	8	188.26	0.00	99.15	1327.73	8	27.41	0.05	98.86	1174.42	8	12.70	3.79	99.98	1173.25	8	12.58	2.15	99.91
7	819.56	10	1244.28	10	51.82	0.00	96.32	994.47	10	21.34	0.02	96.32	874.78	10	6.74	3.03	96.31	**866.60**	10	5.74	2.04	96.62
Avg.	–	–	–	–	69.70	0.00	98.34	–	–	12.62	0.04	99.03	–	–	4.28	4.48	99.09	–	–	4.07	2.28	99.24

Table 5
Results obtained with the Clarke and Wright-based Strategy 1 for the instances based on Christofides et al. (1979).

Ins.	CW + GR					(CW + LS) + GR					(CW + SA) + GR					(CW + RR) + GR				
	OF	No. Rou.	Dev.	Time	Max Den.	OF	No. Rou.	Dev.	Time	Max Den.	OF	No. Rou.	Dev.	Time	Max Den.	OF	No. Rou.	Dev.	Time	Max Den.
1	659.40	8	23.62	0.41	73.40	647.91	8	21.47	0.54	73.40	617.16	8	15.70	52.24	73.13	*614.36*	8	15.18	32.88	73.34
2	1008.78	14	18.63	0.58	77.83	996.99	14	17.24	0.81	78.75	974.09	14	14.55	107.63	77.03	***958.65***	14	12.73	63.41	78.29
3	1060.57	11	27.77	0.66	69.88	1052.51	11	26.80	1.23	69.89	966.46	11	16.43	179.22	71.35	*965.82*	11	16.35	112.40	70.74
4	1309.00	16	25.57	1.11	73.13	1332.04	17	27.78	2.15	70.99	***1295.98***	17	24.32	369.73	69.84	1297.93	17	24.50	201.16	69.38
5	1719.37	24	28.50	1.69	70.87	1755.59	25	31.20	3.67	68.18	*1648.90*	24	23.23	598.95	71.40	1658.81	24	23.97	223.24	70.54
6	1505.35	11	41.56	0.82	69.38	1488.59	11	39.98	1.88	67.69	1466.29	11	37.88	214.38	67.97	***1455.34***	11	36.85	113.75	68.50
7	1215.08	15	40.21	0.67	70.42	1179.82	15	36.14	1.23	70.74	1128.05	14	30.17	159.69	71.36	*1124.66*	14	29.78	85.64	71.16
Avg.	–	–	29.41	0.85	72.13	–	–	28.66	1.64	71.38	–	–	23.18	240.26	71.73	–	–	22.77	118.93	71.71

Table 6
Results obtained with the Gillett and Miller-based Strategy 1 for the instances based on Christofides et al. (1979).

Ins.	GM + GR					(GM + LS) + GR					(GM + SA) + GR					(GM + RR) + GR				
	OF	No. Rou.	Dev.	Time	Max Den.	OF	No. Rou.	Dev.	Time	Max Den.	OF	No. Rou.	Dev.	Time	Max Den.	OF	No. Rou.	Dev.	Time	Max Den.
1	840.61	7	57.59	0.33	76.23	617.30	7	15.73	0.62	75.88	614.12	8	15.13	51.81	73.14	***610.87***	7	14.52	33.90	74.72
2	1170.02	13	37.59	0.41	81.39	1053.59	15	23.90	0.98	76.44	987.24	15	16.10	111.33	77.21	*967.82*	14	13.81	61.76	77.13
3	1343.12	12	61.81	0.37	71.50	1010.64	11	21.75	1.52	70.83	963.67	11	16.09	185.91	72.37	***962.07***	11	15.90	126.97	72.81
4	1861.71	18	78.58	0.45	71.74	1347.19	18	29.23	3.35	71.74	1310.37	19	25.70	369.70	69.28	*1307.48*	17	25.42	189.22	68.72
5	2441.65	24	82.48	0.56	72.61	1811.90	25	35.41	5.53	71.03	1667.80	25	24.64	607.51	70.70	***1646.81***	24	23.07	299.70	70.22
6	3189.82	11	199.96	0.39	70.76	1719.13	11	61.66	2.85	66.36	*1543.58*	11	45.15	203.18	68.47	1545.58	11	45.34	130.25	67.62
7	1375.18	12	58.69	0.38	81.33	1169.48	14	34.95	1.57	71.26	1126.38	14	29.98	160.14	71.36	***1118.70***	13	29.09	112.02	71.36
Avg.	–	–	82.39	0.41	75.08	–	–	31.80	2.35	71.93	–	–	24.68	241.37	71.79	–	–	23.88	136.26	71.80

Table 7
Results obtained with the Clarke and Wright-based Strategy 2 for the instances based on Christofides et al. (1979).

Ins.	CW + GR					(CW + GR) + LS					(CW + GR) + SA					(CW + GR) + RR				
	OF	No. Rou.	Dev.	Time	Max Den.	OF	No. Rou.	Dev.	Time	Max Den.	OF	No. Rou.	Dev.	Time	Max Den.	OF	No. Rou.	Dev.	Time	Max Den.
1	636.25	7	19.28	0.06	77.41	620.98	7	16.42	4.78	77.41	601.01	7	12.68	1155.09	79.62	***595.44****	7	11.63	930.28	77.69
2	1003.99	13	18.07	0.10	79.94	1001.55	13	17.78	8.23	79.94	***959.86***	13	12.88	2392.17	81.94	963.97	13	13.36	1865.33	80.75
3	1013.33	11	22.08	0.13	77.24	1001.33	11	20.63	23.84	77.24	*942.89*	11	13.59	4510.44	76.20	949.02	11	14.33	2847.31	74.11
4	1288.88	16	23.64	0.22	76.26	1279.09	16	22.70	52.09	76.26	1242.67	16	19.20	9751.78	76.09	***1223.45****	16	17.36	5480.90	78.75
5	1659.05	22	23.99	0.38	78.69	1652.31	22	23.48	68.37	78.69	1574.26	22	17.65	18249.83	78.84	***1567.67***	22	17.16	7293.93	78.84
6	1458.65	10	37.17	0.28	74.50	1444.44	10	35.83	37.38	75.08	***1421.68****	10	33.69	6793.13	76.77	1434.43	10	34.89	3773.92	73.89
7	1134.32	13	30.89	0.13	75.67	1115.47	13	28.72	21.11	75.88	*1055.29*	13	21.77	3649.52	80.91	1055.48	13	21.80	2080.32	80.91
Avg.	–	–	25.02	0.19	77.10	–	–	23.65	30.83	77.21	–	–	18.78	6643.14	78.62	–	–	18.65	3698.61	77.85

Table 8
Results obtained with the Gillett and Miller-based Strategy 2 for the instances based on Christofides et al. (1979).

Ins.	GM + GR					(GM + GR) + LS					(GM + GR) + SA					(GM + GR) + RR				
	OF	No. Rou.	Dev.	Time	Max Den.	OF	No. Rou.	Dev.	Time	Max Den.	OF	No. Rou.	Dev.	Time	Max Den.	OF	No. Rou.	Dev.	Time	Max Den.
1	689.94	7	29.35	0.04	77.67	619.88	7	16.21	7.56	76.99	*602.15*	7	12.89	1143.08	77.12	609.24	7	14.22	762.94	77.69
2	1252.89	13	47.33	0.08	80.78	1045.68	13	22.97	24.18	80.63	980.93	13	15.35	2384.86	81.39	*971.78*	13	14.28	3147.63	79.29
3	1391.65	11	67.65	0.09	75.97	1003.97	11	20.95	52.33	74.92	***932.18****	11	12.30	4669.83	76.20	943.10	11	13.62	4805.44	76.06
4	1838.56	16	76.36	0.14	77.76	1305.57	16	25.24	88.55	76.37	*1231.52*	16	18.13	10668.92	79.55	1235.94	16	18.56	5612.73	78.94
5	2389.52	23	78.58	0.16	76.44	1728.05	23	29.14	183.31	76.39	*1580.44*	23	18.11	17331.73	78.84	1589.47	23	18.79	6738.40	78.70
6	3045.62	10	186.40	0.10	72.08	1813.33	10	70.52	101.85	73.32	1507.10	10	41.72	7172.39	76.18	*1422.91*	10	33.81	7020.55	76.18
7	1442.64	14	66.47	0.09	75.88	1149.47	14	32.64	43.38	75.99	1069.64	14	23.43	3668.18	80.91	***1050.92****	13	21.27	3931.22	80.91
Avg.	–	–	78.88	0.10	76.65	–	–	31.10	71.59	76.37	–	–	20.28	6719.86	78.60	–	–	19.22	4574.13	78.25

Table 9
Results obtained with the Clarke and Wright-based Strategy 3 for the instances based on Christofides et al. (1979).

Ins.	[(CW + SA) + GR] + SA					[(CW + RR) + GR] + RR				
	OF	No. Rou.	Dev.	Time	Max Den.	OF	No. Rou.	Dev.	Time	Max Den.
1	***596.04***	8	11.74	1094.76	77.69	613.31	8	14.98	822.84	77.69
2	959.82	14	12.87	2533.43	78.29	***957.65****	14	12.62	2799.50	80.75
3	946.53	11	14.03	4946.01	78.26	***936.55***	11	12.83	2550.28	77.79
4	1239.34	17	18.88	10696.02	79.55	*1237.78*	17	18.73	6076.35	78.25
5	1588.63	24	18.73	17105.95	78.80	*1580.90*	22	18.15	8252.41	80.21
6	1428.43	11	34.32	6977.80	75.28	***1427.59***	10	34.25	4428.19	74.90
7	1072.43	14	23.75	3921.44	75.98	***1057.78***	13	22.06	2297.19	75.98
Avg.	–	–	19.19	6753.63	77.69	–	–	19.09	3889.54	77.94

Table 10
Results obtained with the Gillett and Miller-based Strategy 3 for the instances based on Christofides et al. (1979).

Ins.	[(GM + SA) + GR] + SA					[(GM + RR) + GR] + RR				
	OF	No. Rou.	Dev.	Time	Max Den.	OF	No. Rou.	Dev.	Time	Max Den.
1	***596.04***	8	11.74	1091.67	77.69	603.69	7	13.18	1153.94	77.11
2	973.88	15	14.52	2449.31	78.29	*967.82*	14	13.81	2225.35	77.13
3	*946.64*	11	14.04	4665.56	77.05	948.89	11	14.31	3223.59	74.37
4	1257.90	19	20.66	10198.14	79.55	***1233.33***	16	18.31	5676.53	78.95
5	1590.69	25	18.88	17410.00	78.45	***1567.52****	23	17.15	8963.77	78.45
6	1484.92	11	39.64	6813.68	75.25	*1434.63*	10	34.91	4675.54	78.34
7	1069.07	14	23.36	3863.37	80.91	*1058.48*	13	22.14	4589.50	75.98
Avg.	–	–	20.41	6641.68	78.17	–	–	19.12	4358.32	77.19

Gillett and Miller algorithm was able to find the best solution for 4 out of 7 instances, although the Record-to-Record Travel algorithm building on the Clarke and Wright algorithm obtained the lowest average deviation with 22.77%. Remember that these deviations are obtained with relation to the best solution (among all algorithms proposed) obtained without considering three-dimensional loading constraints, as explained before. In the case of Strategy 2 (see Tables 7 and 8, highlighted in bold), the Record-to-Record Travel algorithm building on the Clarke and Wright algorithm was able to find the best solution for 3 out of 7 instances and it also obtained the lowest average deviation with 18.65%. Finally in the case of Strategy 3 (see Tables 9 and 10, highlighted in bold), the Record-to-Record Travel algorithm building on the Clarke and Wright algorithm was able to find the best solution for 4 out of 7 instances and it also obtained the lowest average deviation with 19.09%.

For this set of instances, considering all strategies (sum of columns "Intra" in Table 11, values highlighted in bold in Tables 5–10), the algorithm that had the best performance was the Record-to-Record Travel algorithm building on the Clarke and Wright algorithm, being able to find 9 best solutions (regarding 7

Table 11
Overview of the results obtained for the instances based on Christofides et al. (1979).

Algorithm based on		Strategy						Total Intra
		1		2		3		
		Intra	Inter	Intra	Inter	Intra	Inter	
Clarke and Wright	Simulated Annealing	1	0	2	1	1	0	4
	Record-to-Record Travel	2	0	3	2	4	1	9
Gillett and Miller	Simulated Annealing	0	0	1	1	1	0	2
	Record-to-Record Travel	4	0	1	1	2	1	7
Total inter		–	0	–	5	–	2	–

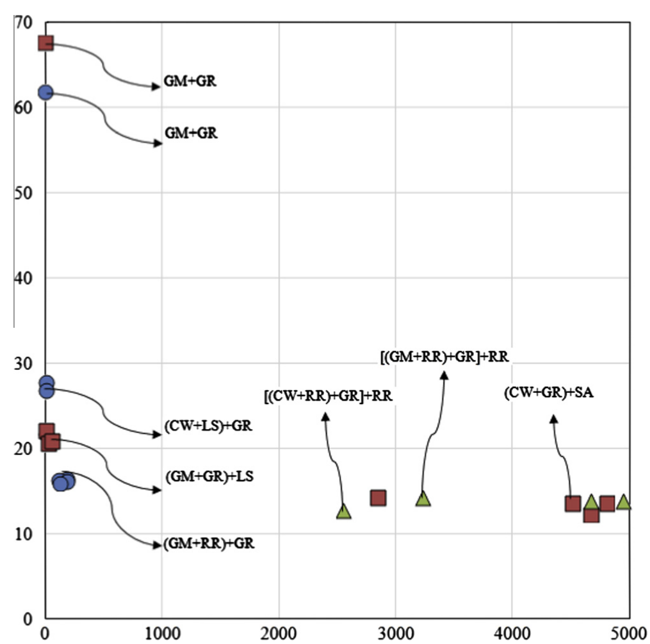

Fig. 9. Overall solution quality × time trade-off for instance 3 of the set based on Christofides et al. (1979).

Table 12
Instances based on the carrier data.

Ins.	No. Cus.	No. Typ.	No. Box.
1	15	4	1720
2	29	8	8024
3	36	7	1683
4	57	3	20,405
5	73	2	3916
6	86	5	10,696
7	97	6	29,580

Table 13
Results obtained with the Clarke and Wright-based algorithms with no three-dimensional loading constraint for the instances based on the carrier data.

Ins.	Best Sol.		CW					CW + LS					CW + SA					CW + RR				
	OF	No. Rou.	OF	No. Rou.	Gap	Time	Max Den.	OF	No. Rou.	Gap	Time	Max Den.	OF	No. Rou.	Gap	Time	Max Den.	OF	No. Rou.	Gap	Time	Max Den.
1	155.03	3	151.28	3	−2.42	0.01	94.25	151.28	3	−2.42	0.02	94.25	**134.83**	3	−13.03	0.14	94.25	135.84	3	−12.38	0.09	98.07
2	203.39	5	246.35	5	21.12	0.01	98.56	246.35	5	21.12	0.01	98.56	**217.63**	5	7.00	0.33	98.56	**217.63**	5	7.00	0.30	98.56
3	199.38	4	247.86	4	24.32	0.01	98.48	246.07	4	23.42	0.01	90.91	**215.99**	4	8.33	0.44	98.48	216.06	4	8.37	0.37	98.48
4	259.49	7	359.16	7	38.41	0.02	99.14	356.92	7	37.55	0.02	99.14	308.76	7	18.99	0.78	99.14	307.40	7	18.26	0.47	99.14
5	327.68	8	443.56	8	35.36	0.02	99.58	441.88	8	34.85	0.02	99.58	373.21	8	13.89	1.14	99.58	372.33	8	13.63	1.11	99.58
6	371.45	10	531.20	10	43.01	0.02	98.80	528.30	10	42.23	0.02	98.80	457.12	10	23.06	1.42	98.80	**447.37**	10	20.44	1.00	98.80
7	590.47	19	770.32	19	30.46	0.02	97.54	770.01	19	30.41	0.03	97.54	691.95	19	17.19	1.64	97.54	**688.02**	19	16.52	1.12	97.54
Avg.	–	–	–	–	27.29	0.02	98.05	–	–	26.85	0.02	96.97	–	–	10.79	0.84	98.05	–	–	10.29	0.64	98.60

Table 14
Results obtained with the Gillett and Miller-based algorithms with no three-dimensional loading constraint for the instances based on the carrier data.

Ins.	Best Sol.		GM					GM + LS					GM + SA					GM + RR				
	OF	No. Rou.	OF	No. Rou.	Gap	Time	Max Den.	OF	No. Rou.	Gap	Time	Max Den.	OF	No. Rou.	Gap	Time	Max Den.	OF	No. Rou.	Gap	Time	Max Den.
1	155.03	3	184.77	3	19.18	0.01	94.25	148.86	3	−3.98	0.01	94.25	**134.83**	3	−13.03	0.16	94.25	**134.83**	3	−13.03	0.12	94.25
2	203.39	5	333.68	5	64.06	0.01	98.56	259.44	5	27.56	0.01	95.22	**217.63**	5	7.00	0.34	98.56	**217.63**	5	7.00	0.34	98.56
3	199.38	4	349.39	5	75.24	0.01	98.48	274.29	5	37.57	0.01	98.48	226.22	5	13.46	0.42	98.48	226.22	5	13.46	0.56	98.48
4	259.49	7	463.12	7	78.47	0.01	99.14	358.67	7	38.22	0.02	99.14	311.47	7	20.03	0.80	98.07	**306.36**	7	18.06	0.61	99.14
5	327.68	8	582.53	9	77.77	0.02	98.81	455.46	9	39.00	0.03	98.81	385.32	9	17.59	1.12	99.58	**368.47**	8	12.45	1.12	99.58
6	371.45	10	757.35	11	103.89	0.01	98.80	551.66	11	48.52	0.03	98.80	457.64	11	23.20	1.47	98.80	449.06	10	20.89	1.31	98.80
7	590.47	19	879.32	20	48.92	0.02	97.54	814.59	20	37.96	0.03	97.54	704.96	20	19.39	1.62	97.54	691.48	19	17.11	1.04	97.54
Avg.	–	–	–	–	68.05	0.01	97.94	–	–	32.51	0.02	97.46	–	–	12.41	0.85	97.90	–	–	10.95	0.73	98.05

Table 15
Results obtained with the Clarke and Wright-based Strategy 1 for the instances based on the carrier data.

Ins.	CW + GR					(CW + LS) + GR					(CW + SA) + GR					(CW + RR) + GR				
	OF	No. Rou.	Dev.	Time	Max Den.	OF	No. Rou.	Dev.	Time	Max Den.	OF	No. Rou.	Dev.	Time	Max Den.	OF	No. Rou.	Dev.	Time	Max Den.
1	154.44	3	14.54	0.30	78.96	154.44	3	14.54	0.31	78.96	***143.37***[*]	3	6.33	5.26	80.24	143.44	3	6.39	3.98	78.96
2	266.36	5	22.39	0.50	81.86	261.74	5	20.27	0.55	81.86	237.76	5	9.25	13.40	80.19	*236.12*	5	8.49	13.04	83.53
3	297.12	6	37.56	0.31	73.23	295.03	6	36.60	0.42	73.23	261.52	6	21.08	17.60	73.23	***260.35***	6	20.54	18.75	75.76
4	382.68	8	24.91	0.64	88.48	376.45	8	22.88	0.84	88.48	***326.16***	8	6.46	36.47	89.54	329.95	8	7.70	30.28	88.48
5	506.53	11	37.47	0.37	77.97	505.49	11	37.19	0.61	77.97	*440.54*	11	19.56	52.28	77.97	440.77	11	19.62	59.58	77.97
6	603.21	13	34.83	0.64	79.04	599.94	13	34.10	1.11	79.04	*521.98*	13	16.68	69.08	79.04	526.75	13	17.74	46.74	79.04
7	854.61	22	24.21	1.26	85.35	852.88	22	23.96	1.64	85.35	*763.17*	22	10.92	80.14	85.35	766.10	22	11.35	61.00	85.35
Avg.	–	–	27.99	0.57	80.70	–	–	27.08	0.78	80.70	–	–	12.90	39.18	80.79	–	–	13.12	33.34	81.30

Table 16
Results obtained with the Gillett and Miller-based Strategy 1 for the instances based on the carrier data.

Ins.	GM + GR					(GM + LS) + GR					(GM + SA) + GR					(GM + RR) + GR				
	OF	No. Rou.	Dev.	Time	Max Den.	OF	No. Rou.	Dev.	Time	Max Den.	OF	No. Rou.	Dev.	Time	Max Den.	OF	No. Rou.	Dev.	Time	Max Den.
1	178.07	3	32.07	0.22	75.14	158.50	3	17.55	0.25	82.78	***143.37***[*]	3	6.33	5.37	80.24	***143.37***[*]	3	6.33	4.21	80.24
2	293.78	5	34.99	0.47	86.87	262.00	5	20.39	0.59	83.53	***233.52***	5	7.30	13.18	85.20	235.22	5	8.08	13.09	81.86
3	341.00	6	57.88	0.28	70.71	293.77	6	36.01	0.47	78.28	*261.52*	6	21.08	17.95	73.23	271.53	6	25.71	24.62	70.71
4	428.41	8	39.84	0.53	89.54	391.52	8	27.80	1.00	90.61	326.50	8	6.58	37.24	89.54	*326.24*	8	6.49	37.58	89.54
5	611.02	11	65.83	0.28	79.51	492.33	10	33.61	0.91	81.83	436.01	11	18.33	52.06	79.51	***423.25***	10	14.87	58.64	79.51
6	699.67	13	56.40	0.47	79.04	604.09	13	35.03	1.34	79.04	522.31	13	16.75	69.17	79.04	***519.30***	13	16.08	58.81	79.04
7	952.08	23	38.38	1.06	85.35	883.95	23	28.48	1.81	85.35	775.14	23	12.66	81.07	85.35	***757.12***	21	10.04	78.28	85.35
Avg.	–	–	46.48	0.47	80.88	–	–	28.41	0.91	83.06	–	–	12.72	39.43	81.73	–	–	12.52	39.32	80.89

Table 17
Results obtained with the Clarke and Wright-based Strategy 2 for the instances based on the carrier data.

Ins.	CW + GR					(CW + GR) + LS					(CW + GR) + SA					(CW + GR) + RR				
	OF	No. Rou.	Dev.	Time	Max Den.	OF	No. Rou.	Dev.	Time	Max Den.	OF	No. Rou.	Dev.	Time	Max Den.	OF	No. Rou.	Dev.	Time	Max Den.
1	154.44	3	14.54	0.02	78.96	154.44	3	14.54	0.23	78.96	***143.37****	3	6.33	78.86	80.24	143.44	3	6.39	58.22	78.96
2	266.52	5	22.47	0.08	83.53	258.34	5	18.70	6.99	83.53	***229.51***	5	5.46	658.68	88.54	234.68	5	7.83	794.50	85.20
3	283.09	5	31.07	0.06	80.81	282.74	5	30.90	2.58	80.81	***254.09****	5	17.64	772.13	80.81	***254.09****	5	17.64	590.49	80.81
4	382.59	8	24.88	0.08	91.68	376.37	8	22.85	15.32	91.68	*324.67*	8	5.98	1670.58	90.61	325.73	8	6.32	1438.62	90.61
5	492.08	10	33.55	0.08	83.37	491.24	10	33.32	6.02	83.37	426.98	10	15.88	879.78	84.14	***418.00****	10	13.44	1326.28	85.69
6	581.66	12	30.02	0.12	83.98	581.24	12	29.92	14.68	83.98	505.27	12	12.94	2977.86	83.98	***504.13****	12	12.69	2667.98	83.98
7	854.61	22	24.21	0.31	85.35	852.88	22	23.96	16.43	85.35	767.65	22	11.57	2516.49	85.35	*753.26*	21	9.48	1575.51	91.44
Avg.	–	–	25.82	0.11	83.95	–	–	24.89	8.89	83.95	–	–	10.83	1364.91	84.81	–	–	10.54	1207.37	85.24

Table 18
Results obtained with the Gillett and Miller-based Strategy 2 for the instances based on the carrier data.

Ins.	GM + GR					(GM + GR) + LS					(GM + GR) + SA					(GM + GR) + RR				
	OF	No. Rou.	Dev.	Time	Max Den.	OF	No. Rou.	Dev.	Time	Max Den.	OF	No. Rou.	Dev.	Time	Max Den.	OF	No. Rou.	Dev.	Time	Max Den.
1	189.53	3	40.57	0.01	85.33	175.66	3	30.29	0.72	80.24	***143.37****	3	6.33	73.37	80.24	143.44	3	6.39	62.40	78.96
2	293.78	5	34.99	0.06	86.87	272.99	5	25.44	7.24	86.87	***229.51***	5	5.46	661.18	88.54	234.68	5	7.83	667.35	85.20
3	377.90	6	74.96	0.04	78.28	303.94	6	40.72	4.63	73.23	260.32	6	20.53	431.04	78.28	*259.81*	6	20.29	539.41	78.28
4	487.05	8	58.98	0.09	84.21	394.40	8	28.74	21.78	90.61	327.39	8	6.86	1682.93	90.61	***314.55***	7	2.67	2072.67	90.61
5	609.53	10	65.42	0.08	85.69	500.17	10	35.74	12.09	83.37	427.02	10	15.89	872.40	81.83	*419.95*	10	13.97	916.75	82.60
6	684.20	12	52.94	0.08	83.98	599.62	12	34.03	48.33	83.98	*505.78*	12	13.06	2991.67	83.98	505.82	12	13.07	1826.53	83.98
7	954.95	23	38.80	0.16	85.35	895.77	23	30.20	34.68	85.35	781.50	23	13.59	2392.75	91.44	***752.94***	21	9.44	2294.51	91.44
Avg.	–	–	52.38	0.07	84.24	–	–	32.16	18.50	83.38	–	–	11.67	1300.76	84.99	–	–	10.52	1197.09	84.44

Table 19
Results obtained with the Clarke and Wright-based Strategy 3 for the instances based on the carrier data.

Ins.	[(CW + SA) + GR] + SA					[(CW + RR) + GR] + RR				
	OF	No. Rou.	Dev.	Time	Max Den.	OF	No. Rou.	Dev.	Time	Max Den.
1	***143.37****	3	6.33	86.78	80.24	143.44	3	6.39	61.61	78.96
2	*229.51*	5	5.46	734.50	88.54	235.72	5	8.31	788.70	88.54
3	***255.70***	6	18.39	484.13	78.28	***255.70***	6	18.39	485.33	75.76
4	326.16	8	6.46	1822.91	89.54	*324.93*	8	6.06	2815.74	90.61
5	439.64	11	19.31	857.47	85.69	*421.09*	10	14.28	736.84	84.14
6	510.67	13	14.15	2901.80	83.98	*505.61*	12	13.02	1516.68	83.98
7	762.92	22	10.89	2735.80	85.35	*754.56*	21	9.67	1462.32	91.44
Avg.	–	–	11.57	1374.77	84.52	–	–	10.87	1123.89	84.78

Table 20
Results obtained with the Gillett and Miller-based Strategy 3 for the instances based on the carrier data.

Ins.	[(GM + SA) + GR] + SA					[(GM + RR) + GR] + RR				
	OF	No. Rou.	Dev.	Time	Max Den.	OF	No. Rou.	Dev.	Time	Max Den.
1	***143.37****	3	6.33	87.18	80.24	***143.37****	3	6.33	67.15	80.24
2	***228.57****	5	5.03	721.96	88.54	234.82	5	7.90	734.55	88.54
3	***255.70***	6	18.39	17.95	78.28	259.93	6	20.34	774.82	78.28
4	326.50	8	6.58	1796.04	89.54	***313.45****	7	2.31	2291.86	90.61
5	436.01	11	18.33	847.82	79.51	***420.37***	10	14.09	1100.90	84.91
6	513.80	13	14.85	2960.58	83.98	***505.55***	12	13.00	3459.99	83.98
7	775.14	23	12.66	2446.58	85.35	***751.40****	21	9.21	3240.55	91.44
Avg.	–	–	11.74	1268.30	83.63	–	–	10.46	1667.12	85.43

Table 21
Overview of the results obtained for the instances based on the carrier data.

Algorithm based on		Strategy						Total Intra
		1		2		3		
		Intra	Inter	Intra	Inter	Intra	Inter	
Clarke and Wright	Simulated Annealing	2	1	3	2	2	1	7
	Record-to-Record Travel	1	0	3	3	1	0	5
Gillett and Miller	Simulated Annealing	2	1	2	1	3	2	7
	Record-to-Record Travel	4	1	2	0	5	3	11
Total inter	–	–	3	–	6	–	6	–

out of 7 instances), followed by the Record-to-Record Travel algorithm building on the Gillett and Miller algorithm, being able to find 7 best solutions (regarding 5 out of 7 instances). On the other hand, considering all algorithms (sum of lines "Inter" in Table 11, values marked with an asterisk (*) in Tables 5–10), the strategy that had the best performance was the Strategy 2, being able to find 5 best solutions (regarding 5 out of 7 instances), followed by the Strategy 3, being able to find 2 best solutions (regarding 2 out of 7 instances).

In the case of Strategy 1, it is still interesting to note that a better method for the CVRP not necessarily leads to better solutions for the 3L-CVRP. It occurs with the Clarke and Wright algorithm with the Local Search operators for instances 4 and 5 (see Table 5), whose objective function values are worse than the ones obtained with the Clarke and Wright algorithm (with no improvement). It is due to the nature of this strategy, once, when varying the vehicles capacity, different methods may generate different routes, and therefore they may follow through different paths during the construction of a solution for the 3L-CVRP (see Fig. 7). It is also interesting to highlight the differences of the computational times between Strategies 1 and 2, being these times substantially greater on the latter, which is due to the large number of evaluations performed to verify the feasibility of the moves in the steps that attempted to improve the solutions quality. It is also interesting to note the lower maximum density values obtained for all instances in Tables 5–10, whose fall with relation to Tables 3 and 4 is due to the impact of the three-dimensional loading constraints.

Lastly, Fig. 9 shows a general solution quality (vertical axis) $\times$ time (horizontal axis) trade-off for all employed algorithms and strategies on instance 3 of this data set, where the circles denote Strategy 1, the squares denote Strategy 2 and the triangles denote Strategy 3. Some of the algorithms have their references also illustrated. The usefulness of this trade-off analysis is to help the decision maker to choose which algorithm/strategy would be more adequate to be applied on a practical context.

5.2. Experiments on instances based on a real situation

In a second set of experiments it was considered seven instances obtained with the carrier (see Section 2). The instances consider symmetric costs with the number of nodes n varying from 16 to 98. The coordinates of node 1 (i.e., the depot) and the coordinates of the remaining nodes (i.e., the customers) are given in terms of a Geographic coordinate system (i.e., latitudes and longitudes). The depot is not necessarily centered with relation to the customers. The number of box types considered in these instances varies from 2 to 8 with dimensions in millimeters, and there are no repeated box types among the instances. The demand values (in terms of volume) of the customers for each of these box types were also obtained from the carrier. The total number of boxes varies from 1683 to 29,580. Table 12 presents, for each instance (Ins.), the number of customers (No. Cus.), the number of box types (No. Typ.) and the total number of boxes (No. Box.).

Tables 13 and 14 also present, for each instance, the objective function value (OF) and the number of routes/vehicles (No. Rou.) present in the solution reported by the carrier, which uses the software Roadnet® (see Section 2). That is, without considering the arrangement of the cargo inside the vehicles and assuming therefore that the stowage loss value is equal to zero. It implies that the solutions obtained by Roadnet® for these instances have no guarantees to be feasible from the point of view of the arrangement of the cargo (and they are probably infeasible). The computational times for the Roadnet® to obtain these solutions were not reported by the company, as well as the specifications of the hardware used. Note that the gaps (Gap, in%) in Tables 13 and 14 are obtained with relation to the solution reported by the carrier, while the deviations (Dev., in%) in Tables 15–20 are obtained with relation to the best solution (among all algorithms proposed) obtained without considering three-dimensional loading constraints (see Tables 13 and 14, values highlighted in bold). Both values are computed similarly as follows:

$$\frac{(\text{best obtained} - \text{best carrier})}{(\text{best carrier})}100\%$$

$$\text{or } \frac{(\text{best obtained with 3D} - \text{best obtained without 3D})}{(\text{best obtained without 3D})}100\%$$

For this set of instances, in terms of solution quality, the algorithm that had the best performance was the Record-to-Record Travel algorithm building on the Clarke and Wright algorithm, having obtained an average gap of 10.29% in an average computational time of 0.64 s. Then appears the Simulated Annealing algorithm building on the Clarke and Wright algorithm, having obtained an average gap of 10.79% in an average computational time of 0.84 s. It is also interesting to note the high values of maximum density that were obtained for all instances, which is an evidence that these solutions are probably not feasible from the point of view of the arrangement of the cargo.

In the case of Strategy 1 (see Tables 15 and 16, highlighted in bold), the Record-to-Record Travel algorithm building on the Gillett and Miller algorithm was able to find the best solution for 4 out of 7 instances and it also obtained the lowest average deviation with 12.52%. Remember that these deviations are obtained with relation to the best solution (among all algorithms proposed) obtained without considering three-dimensional loading constraints, as explained before. In the case of Strategy 2 (see Tables 17 and 18, highlighted in bold), the Simulated Annealing and the Record-to-Record Travel algorithms, both building on the Clarke and Wright algorithm, were able to find the best solution for 3 out of 7 instances, although the Record-to-Record Travel algorithm building on the Gillett and Miller algorithm obtained the lowest average deviation with 10.52%. Finally in the case of Strategy 3 (see Tables 19 and 20, highlighted in bold), the Record-to-Record Travel algorithm building on the Gillett and Miller algorithm was able to find the best solution for 5 out of 7 instances and it also obtained the lowest average deviation with 10.46%.

For this set of instances, considering all strategies (sum of columns "Intra" in Table 21, values highlighted in bold in Tables 15–20), the algorithm that had the best performance was the Record-to-Record Travel algorithm building on the Gillett and Miller algorithm, being able to find 11 best solutions (regarding 5 out of 7 instances), followed by the Simulated Annealing algorithm building on the Clarke and Wright algorithm, being able to find 7 best solutions (regarding 4 out of 7 instances), and by the Simulated Annealing algorithm building on the Gillett and Miller algorithm, being able to find 7 best solutions (regarding 3 out of 7 instances). On the other hand, considering all algorithms (sum of lines "Inter" in Table 21, values marked with an asterisk (*) in Tables 15–20), the strategies that had the best performance were the Strategies 2 and 3, being able to find 6 best solutions each (regarding 4 out of 7 instances).

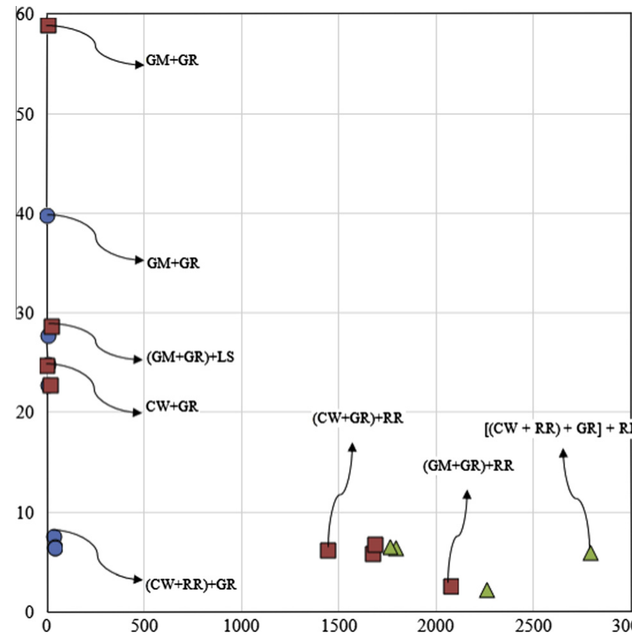

Fig. 10. Overall solution quality × time trade-off for instance 4 of the set based on a real situation.

In the case of the experiments with algorithms considering no three-dimensional loading constraint, it is still interesting to note the presence of some negative values for the Gap. It occurs with all algorithms for instance 1, except the Gillett and Miller algorithm (with not improvement) (see Table 14). It means that, for this instance only, the solutions obtained by these methods are better than the ones reported by the carrier. It is once more interesting to highlight the differences of the computational times between Strategies 1 and 2, being these times substantially greater on the latter, which is due to the large number of evaluations performed to verify the feasibility of the moves in the steps that attempted to improve the solutions quality. It is also interesting to note the lower maximum density values obtained for all instances in Tables 15–20, whose fall with relation to Tables 13 and 14 is due to the impact of the three-dimensional loading constraints.

Lastly, Fig. 10 shows a general solution quality (vertical axis) × time (horizontal axis) trade-off for all employed algorithms and strategies on instance 4 of this data set, where the circles denote Strategy 1, the squares denote Strategy 2 and the triangles denote Strategy 3. Some of the algorithms have their references also illustrated. Once more, the usefulness of this trade-off analysis is to help the decision maker to choose which algorithm/strategy would be more adequate to be applied on a practical context.

6. Conclusion

In this paper, we present reasonably simple and effective heuristic algorithms for a three-dimensional loading capacitated vehicle routing problem arising in a Brazilian carrier. It is assumed that the boxes and the containers/vehicles are of rectangular shape. The objective is to find minimum cost delivery routes, while ensuring that the boxes are packed completely inside the vehicles and that they do not overlap each other, the vertical stability of the cargo and the multi-drop situations. The algorithms are designed to deal with instances with hundreds of customers and thousands of boxes of relatively few types, which is a common scenario in carrier services (as in the company visited during this study) or when companies need to perform deliveries of batches of goods to warehouses or to retailers, or from warehouses to retailers. The methods can also contribute for a greater assertiveness in shop floor operations, especially in what concerns the cargo arrangement inside the vehicles.

The algorithms are based on the combination of classical heuristics from both vehicle routing and container loading literatures, as well as two metaheuristic strategies, and their use in more elaborate procedures. Although these approaches cannot assure optimal solutions for the respective problems, they are relatively simple, fast enough to solve real instances, flexible enough to include other practical considerations, and normally assure relatively good solutions in acceptable computational times in practice. Most of these attributes are aligned with that advocated by Cordeau et al. (2002), to whom simplicity and flexibility are as important as solution quality and computational time when designing VRP heuristics. The approaches are also sufficiently generic to be embedded with algorithms other than those considered in this study, as well as they can be easily adapted to consider other practical constraints, such as the load bearing strength of the boxes, time windows and pickups and deliveries.

Computational tests were performed with these methods considering instances based on the vehicle routing literature and actual customers' orders, as well as instances based on data collected at the carrier. The results show that the heuristics are able to produce relatively good solutions for real instances with hundreds of customers and thousands of boxes. In the near future we intend to extend these approaches to treat other variants of the problem, such as those involving time windows, pickups and deliveries, among others, and also to embed into the approaches other constraints that are common in freight transportation, such as grouping or separation of boxes inside a vehicle, weight distribution within the vehicle, among others. Another interesting topic for future research is to address situations where boxes need to be packed on pallets first, and then the pallets need to be packed on trucks (Zachariadis et al., 2012, 2013), whereby it is possible to obtain global utilization indices which are useful to evaluate the economical performance of unit load systems in the logistics chain of a company (Morabito, Morales, & Widmer, 2000).

Acknowledgements

The authors would like to thank an anonymous reviewer for the useful comments and suggestions, and Marcos Roberto Silva and Rejane Arinos Vasco of the Brazilian carrier for their valuable collaboration with this study. This research was partially supported by FAPESP (Grants 09/07423-9 and 14/16906-1) and CNPq.

References

Bischoff, E. E., & Ratcliff, M. S. W. (1995). Issues in the development of approaches to container loading. *Omega, 23*(4), 377–390.
Bortfeldt, A. (2012). A hybrid algorithm for the capacitated vehicle routing problem with three-dimensional loading constraints. *Computers and Operations Research, 39*(9), 2248–2257.
Bortfeldt, A., & Homberger, J. (2013). Packing first, routing second – A heuristic for the vehicle routing and loading problem. *Computers and 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.
Cecilio, F. O., & Morabito, R. (2004). Refinamentos na heurística de George e Robinson para o problema de carregamento de caixas dentro de contêineres. *Transportes, 11*(2), 32–45.
Ceschia, S., & Schaerf, A. (2013). Local search for a multi-drop multi-container loading problem. *Journal of Heuristics, 19*(2), 275–294.
Ceschia, S., Schaerf, A., & Stützle, T. (2013). Local search techniques for a routing-packing problem. *Computers and Industrial Engineering, 66*(4), 1138–1149.
Christensen, S. G., & Rousøe, D. M. (2009). Container loading with multi-drop constraints. *International Transactions in Operational Research, 16*(6), 727–743.
Christofides, N., Mingozzi, A., & Toth, P. (1979). The vehicle routing problem. In N. Christofides, A. Mingozzi, P. Toth, & C. Sandi (Eds.), *Combinatorial optimization* (pp. 315–338). Chichester: John Wiley & Sons Ltd..
Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research, 12*(4), 568–581.
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.
Dueck, G. (1993). New optimization heuristics: the great deluge algorithm and the record-to-record travel. *Journal of Computational Physics, 104*(1), 86–92.
Fanslau, T., & Bortfeldt, A. (2010). A tree search algorithm for solving the container loading problem. *INFORMS Journal on Computing, 22*(2), 222–235.
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.
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.
George, J. A., & Robinson, D. F. (1980). A heuristic for packing boxes into a container. *Computers and Operations Research, 7*(3), 147–156.
Gillett, B. E., & Miller, L. R. (1974). A heuristic algorithm for the vehicle-dispatch problem. *Operations Research, 22*(2), 340–349.
Golden, B. L., Raghavan, S., & Wasil, EA. (2008). *The vehicle routing problem: Latest advances and new challenges*. Springer.
Gonçalves, J. F., & Resende, M. G. C. (2012). A parallel multi-population biased random-key genetic algorithm for a container loading problem. *Computers and Operations Research, 39*(2), 179–190.
Groër, C., Golden, B., & Wasil, E. (2010). A library of local search heuristics for the vehicle routing problem. *Mathematical Programming Computation, 2*(2), 79–101.
Iori, M., & Martello, S. (2010). Routing problems with loading constraints. *TOP, 18*(1), 4–27.
Iori, M., & Martello, S. (2013). An annotated bibliography of combined routing and loading problems. *Yugoslav Journal of Operations Research, 23*(3), 311–326.
Junqueira, L., Morabito, R., & Yamashita, D. S. (2012a). Three-dimensional container loading models with cargo stability and load bearing constraints. *Computers and Operations Research, 39*(1), 74–85.
Junqueira, L., Morabito, R., & Yamashita, D. S. (2012b). Mip-based approaches for the container loading problem with multi-drop constraints. *Annals of Operations Research, 199*(1), 51–75.
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.
Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220*(4598), 671–680.
Lacomme, P., Toussaint, H., & Duhamel, C. (2013). A GRASP × ELS for the vehicle routing problem with basic three-dimensional loading constraints. *Engineering Applications of Artificial Intelligence, 26*(8), 1795–1810.
Lai, K. K., Xue, J., & Xu, B. (1998). Container packing in a multi-customer delivering operation. *Computers and Industrial Engineering, 35*(1–2), 323–326.
Laporte, G. (2009). Fifty years of vehicle routing. *Transportation Science, 43*(4), 408–416.
Morabito, R., Morales, S. R., & Widmer, J. A. (2000). Loading optimization of palletized products on trucks. *Transportation Research Part E: Logistics and Transportation Review, 36*(4), 285–296.
Moura, A., & Oliveira, J. F. (2009). An integrated approach to the vehicle routing and container loading problems. *OR Spectrum, 31*(4), 775–800.
Ruan, Q., Zhang, Z., Miao, L., & Shen, H. (2013). A hybrid approach for the vehicle routing problem with three-dimensional loading constraints. *Computers and Operations Research, 40*(6), 1579–1589.
Tao, Y., & Wang, F. (2013). An effective tabu search approach with improved loading algorithms for the 3L-CVRP. *Computers and Operations Research*. http://dx.doi.org/10.1016/j.cor.2013.10.017.
Tarantilis, C. D., Zachariadis, E. E., & Kiranoudis, C. T. (2009). A hybrid metaheuristic algorithm for the integrated vehicle routing and three-dimensional container-loading problem. *IEEE Transactions on Intelligent Transportation Systems, 10*(2), 255–271.
Toth, P., & Vigo, D. (2002). *The vehicle routing problem*. Philadelphia: SIAM Monographs on Discrete Mathematics and Applications.
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. (2012). The pallet-packing vehicle routing problem. *Transportation Science, 46*(3), 341–358.
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.
Zheng, J.-N., Chien, C.-F., & Gen, M. (2014). Multi-objective multi-population biased random-key genetic algorithm for the 3-D container loading problem. *Computers and Industrial Engineering*, 8. http://dx.doi.org/10.1016/j.cie.2014.07.012.
Zhu, W., & Lim, A. (2012). A new iterative-doubling greedy-lookahead algorithm for the single container loading problem. *European Journal of Operational Research, 222*(3), 408–417.
Zhu, W., Qin, H., Lim, A., & Wang, L. (2012). A two-stage tabu search algorithm with enhanced packing heuristics for the 3L-CVRP and M3L-CVRP. *Computers and Operations Research, 39*(9), 2178–2195.