

An ILS-Biased Randomization algorithm for the Two-dimensional Loading HFVRP with Sequential Loading and Items Rotation

Oscar Dominguez¹, Angel A. Juan², Ignacio de la Nuez¹, Djamilia Ouelhadj³

(1) Institute of Intelligent Systems and Numerical Applications in Engineering. University of Las Palmas de Gran Canaria, 35017 Las Palmas de Gran Canaria, Spain. E-mail: oscar@opein.com, didie@ulpgc.es

(2) Computer Science Dept. IN3 - Open University of Catalonia, 08018 Barcelona, Spain. E-mail: ajuanp@uoc.edu

(3) Department of Mathematics. University of Portsmouth, Portsmouth PO1 2UP, UK. E-mail: djamilia.ouelhadj@port.ac.uk

Abstract:

This paper discusses the Two-dimensional Loading Vehicle Routing Problem with Heterogeneous Fleet, Sequential Loading, and Items Rotation (2L-HFVRP-SR). Despite the fact that the 2L-HFVRP-SR can be found in many real-life situations related to the transportation of voluminous items where heterogeneity of fleets, two-dimensional packing restrictions, sequential loading, and items rotation have to be considered, this rich version of vehicle routing-and-packing problem has been rarely analyzed in the literature. Accordingly, this paper contributes to fill this gap by presenting a relatively simple-to-implement algorithm which is able to provide state-of-the-art solutions for such a complex problem in short computation times. The proposed algorithm integrates, inside an Iterated Local Search framework, biased-randomized versions of both vehicle routing and packing heuristics. The efficiency of the proposed algorithm is validated throughout an extensive set of computational tests.

Keywords: Routing; Packing; Heterogeneous Fleet; Iterated Local Search; Biased Randomization

1 Introduction

This paper considers a rich variant of the Vehicle Routing Problem (**Toth and Vigo 2002; Golden et al. 2008; Caceres et al. 2015**) that combines the vehicle routing and loading (packing) decisions. In particular, the paper deals with a quite realistic and new variant of the VRP, known as the Two-dimensional Loading Heterogeneous Fleet Vehicle Routing Problem (2L-HFVRP). This new VRP variant is a generalization of the Two-dimensional Capacitated Vehicle Routing Problem (2L-CVRP), since it relaxes the homogeneous fleet assumption. In the 2L-CVRP, the customers' demands consist of lots of rectangular weighted items which cannot be stacked because of their fragility, weight or shape. Note that the way these items are assigned to vehicles and loaded (packed) into them can have a major influence on the distribution costs. Thus, in addition to the routing issue, decision-makers might also face a Two-Dimensional Packing Problem (**Riff et al. 2009**). Our work was originally motivated by a real-world case at Opein (www.opein.com), a medium-size enterprise which provides industrial equipment to its customers, mostly in the building-construction field. Opein has to periodically deliver a large variety of industrial equipment, including: aerial-work platforms, energy-generation sets, compressors, dumpers,

forklifts, professional cleaning equipment, etc. Similar issues arise in other enterprises which also deliver large-size items to their clients, e.g.: kitchen appliances, furniture, etc. Note that these items, which are assumed for the purposes of this paper to have a rectangular shape, must be efficiently packed on the truck surface in order to attain high levels of vehicle utilization. Therefore, the distribution of such equipment has to be done considering not only their weight but also their specific dimensions in width and length. Usually, such equipment cannot be piled one on top of the other, and cannot overlap either, making this a two-dimensional loading routing problem. From the point of view of the loading problem, there are different possible scenarios: (i) oriented loading, where rotation of items is not allowed –i.e., it is assumed that all items have a fixed orientation given as an input of the problem; (ii) non-oriented loading, where it is allowed to rotate items by 90° during the packing process; (iii) sequential loading, where items are always loaded in reverse order to the order in which customers are visited but re-arrangements of items inside the vehicle are not allowed once the route has started; and (iv) unrestricted loading, where items are allowed to be re-arranged during the distribution process.

In this paper we consider a realistic scenario characterized by the existence of a heterogeneous fleet, sequential loading requirements, and the possibility of rotating the rectangular items by 90° (non-oriented case) during the loading stage. To the best of our knowledge, the 2L-HFVRP has only been analyzed in **Leung et al. (2013)**, and **Dominguez et al. (2014a)**. However, neither of these works considers the more realistic scenario combining the following three characteristics: heterogeneous fleet, allowance of items rotation, and requirement of sequential loading. In our opinion, heterogeneous fleets and items rotation during the packing process are realistic assumptions which have not received enough attention in the literature. Likewise, sequential loading might be a frequent requirement in real-life distribution practices, since unloading and re-loading heavy machinery could represent a significant cost both in terms of time as well as required human resources. **Figure 1** provides an example of sequential loading for a small 3-vehicle case. Notice that the items have been loaded in the truck following an inverse order to the one in which the customers are visited (represented by a number inside a star).

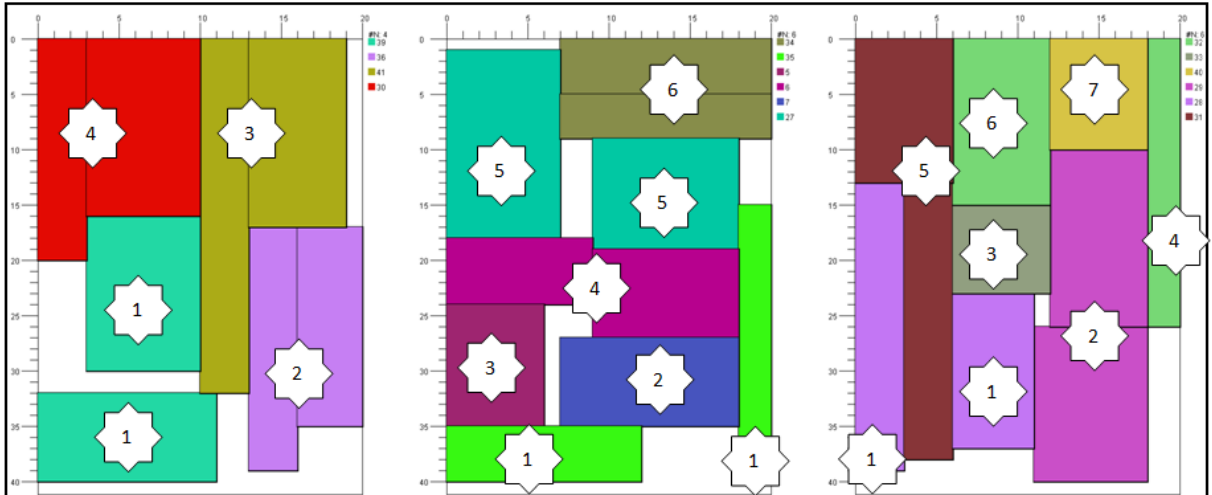


Figure 1: Example of a sequential loading for a small 3-vehicle case.

Therefore, this work proposes an algorithm for solving the Two-dimensional Loading Vehicle Routing Problem with Heterogeneous Fleet, Sequential Loading, and Items Rotation (2L-HFVRP-SR). Our algorithm integrates,

inside an Iterated Local Search (ILS) framework (Lourenço et al. 2003), biased-randomized versions of classical routing and packing heuristics. Biased-randomization of heuristics refers to the use of skewed probability distributions to induce ‘biased’ (non-symmetric) random behavior in a heuristic. As described in Juan et al. (2013a), this technique allows transforming a deterministic heuristic into a multi-start probabilistic algorithm. On the one hand, for the routing part we use a biased-randomized version of the savings heuristic (Clarke and Wright, 1964). This version is also enriched with memory-based and splitting local search strategies, as described in Juan et al. (2011). On the other hand, for the packing part we use biased-randomized versions of the Best-Fit packing heuristic (Burke et al. 2004) and the Touching Perimeter algorithm (Lodi et al. 1999). The packing process is then integrated inside the savings-based routing process.

This paper is structured as follows: Section 2 describes the 2L-HFVRP-SR model in detail. Section 3 reviews some related work. Section 4 gives an overview of our approach, including an explanation on our biased-randomized technique. Section 5 provides low-level details about our approach. Section 6 describes some numerical experiments that contribute to illustrate and validate our approach. Finally, the Conclusions section summarizes the main contributions and results of this work.

2 Problem formulation

The 2L-HFVRP model is based on a complete undirected graph $G = (N, E)$, where $N = \{0, 1, \dots, n\}$ is a set of $n+1$ nodes, representing the central depot (node 0) and the n customers to be supplied (nodes 1 to n), and $E = \{(i, j) / i, j \in N \text{ with } i \neq j\}$ is an edge set. For any pair of nodes (i, j) , with $i \neq j$, there exists an edge e_{ij} with an associated traveling distance $d_{ij} > 0$. Transportation of goods is performed by using a fleet of P different types of vehicles, initially located at the depot. The number of available vehicles of each type is assumed to be unbounded. Each type of vehicle t ($t = 1, 2, \dots, P$) has the following properties: fixed cost F_t , variable cost per unit of distance traveled V_t , maximum weight-loading capacity Q_t and a loading area $A_t = W_t \times L_t$. Furthermore, it is assumed that vehicles with higher weight-loading capacity show higher fixed and variable costs too. Therefore, the fixed cost associated with a route R using a vehicle of type t will be F_t , while its variable cost is

calculated as $VC_R = V_t \cdot \sum_{k=1}^{k=RE} d_{Rk}$, where RE is the number of edges in the route. For each customer i ($i \in N - \{0\}$), $m_i > 0$ is the number of items requested, and D_i is the total weight of these items. It is assumed that the depot has no demand, i.e. $m_0 = 0$. For each customer’s item I_{il} ($1 \leq I_{il} \leq m_i$), its length and width dimensions are denoted by h_{il} and w_{il} , respectively.

Depending on whether or not items can be rotated by 90° during the packing process, and whether or not items are allowed to be re-arranged during the distribution process, Fuellerer et al. (2009) proposed a classification of four different loading configurations: (a) two-dimensional sequential oriented loading (2/SO/L); (b) two-dimensional unrestricted oriented loading (2/UO/L); (c) two-dimensional sequential non-oriented (rotated) loading (2/SR/L); and (d) two-dimensional unrestricted non-oriented (rotated) loading (2/UR/L). This paper deals with the 2L-HFVRP with sequential oriented (2/SO/L) and non-oriented (2/SR/L) loading. In summary, the following routing and packing constraints assumed in this paper are as follows:

- Each route starts and ends at a central depot (closed-routes assumption).
- Each customer cannot be served by more than one vehicle.

- The total load transported by each vehicle cannot surpass the vehicle's maximum weight-loading capacity, Q_t .
- Every item must be loaded with its edges parallel to the edges of the vehicles (orthogonal loading).
- The rectangular items have to be contained within the vehicle loading surface area and without overlapping.
- All items associated to any given customer must be loaded and unloaded from the rear side, employing only straight movements (one per item), so rearrangements of the items in the vehicle at the customers' sites are not allowed (sequential loading constraint).
- Items are allowed to be rotated 90° (non-oriented loading).

Under these assumptions, the goal of the 2L-HFVRP is to find a set of routes satisfying all customers' demands at a minimum cost while not violating any of the aforementioned constraints.

3 Literature review

During the last decade, real-life constraints have enriched the classical Vehicle Routing Problem (VRP). This has led to the emergence of new variants, such as for example the Two-dimensional Loading Heterogeneous Fleet Vehicle Routing Problem (2L-HFVRP). **Leung et al. (2013)** proposed a Simulated Annealing with heuristic local search (SA_HLS) to solve this VRP variant. They also give a set of benchmarks which we will use later to test the efficiency of our approach. However, their paper limits its study to the two-dimensional sequential and unrestricted oriented loading cases, i.e. without allowing rotation of items. **Dominguez et al. (2014a)** used a multi-start biased randomized (MS-BR) algorithm for solving unrestricted two-dimensional loading configurations of the 2L-HFVRP, including the oriented and non-oriented versions. As far as we know, these are the only two papers that have analyzed the 2L-HFVRP.

The 2L-HFVRP integrates two combinatorial optimization problems: the Two-dimensional Capacitated Vehicle Routing Problem (2L-CVRP) and the Heterogeneous Fleet Vehicle Routing Problem (HVRP). For this reason, we will now provide a brief literature review on the 2L-CVRP and the HVRP. The 2L-CVRP was originally introduced by **Iori et al. (2007)**. They proposed an exact branch-and-cut algorithm for the routing aspects, whereas the loading requirements of the problem are addressed through the use of lower bounds, effective heuristics and a specialized branch-and-bound algorithm. However, from the point of view of the loading configuration, they only address the two-dimensional sequential oriented loading case (2|SO|L). Although their approach is able to solve instances with up to 35 customers and more than 100 items, there are smaller instances which remain unsolved. To deal with larger-size 2L-CVRP instances with up to 255 customers and 786 items, **Gendreau et al. (2008)** proposed a Tabu Search algorithm designed to solve both the sequential (2|SO|L) and the unrestricted oriented (2|UO|L) loading configurations. However, the non-oriented (rotated) loading configuration has rarely been addressed. As far as we know, only **Fuellerer et al. (2009)** have solved all four loading configurations. To address this problem, they propose an Ant Colony Optimization (ACO) algorithm. Also, **Dominguez et al. (2014b)** proposed a MultiStart-BiasedRand (MS-BR) algorithm to solve both the two-dimensional unrestricted oriented and the non-oriented loading variants. Nevertheless, the oriented

versions have received much more attention in the 2L-CVRP literature. **Zachariadis et al. (2009)** used a hybrid algorithm combining Tabu Search with a guided local search and five packing heuristics –with different selection criteria– to develop feasible item loadings. **Leung et al. (2011)** proposed an Extended Guided Tabu Search (EGTS) for the main routing problem and add a new heuristic for the load configuration checking. Additionally, **Leung et al. (2010)** employed a Simulated Annealing algorithm to tackle the oriented versions of 2L-CVRP. **Duhamel et al. (2009; 2011)** proposed a GRASP_xELS (Greedy Randomized Adaptive Search Procedure hybridized with Evolutionary Local Search) (**Prins, 2009**). Their approach includes an innovative methodology to address the 2|UO|L version of the 2L-CVRP based on transformation of the 2L problem into a Resource Constrained Project Scheduling Problem. More recently, **Zachariadis et al. (2013)** presented an effective approach named PRMP (Promise Routing-Memory Packing), with a compact structure. Their approach combines a local-search method with an effective diversification based on regional aspiration criteria, which allows them to deal with the routing aspects. The loading feasibility of routes is investigated by a packing heuristic and an innovative simple-structured memory mechanism. They also introduced several memory-based components to reduce the computational effort required to examine the loading feasibility constraint. Another realistic extension of the 2L-CVRP, the two-dimensional loading capacitated vehicle routing problem with time windows (2L-CVRPTW), was introduced by **Khebbache-Hadji et al. (2013)**. Additionally, **Malapert et al. (2008)** proposed a framework to handle the pick-up and delivery extension of the 2L-CVRP. A survey on Vehicle Routing Problem with Loading Constraints can be found in **Wang et al. (2009)** and **Iori and Martello (2010)**.

Unlike the 2L-CVRP, the Heterogeneous Fleet Vehicle Routing Problem (HVRP) has been studied extensively in the VRP literature. According to the number of available vehicles **Baldacci et al. (2008)** proposed a classification of these problems in HFVRP with unlimited fleet –also known as the Fleet Size and Mix (FSM)–, and the limited (HVRP). Furthermore, they consider different costs per type of vehicle, distinguishing between: fixed (F), variable (D), or both (FD). The HFVRP with unlimited fleet (FSM) consists in determining its optimal routing scheme with the best fleet composition. The HFVRP with limited fleet consists in optimizing the use of the available fixed fleet. **Golden et al. (1984)** published one of the first papers analyzing the heterogeneous VRP. In this paper, they consider an unlimited number of vehicles. Since then, different features of heterogeneous VRPs have been studied. Regarding the version with a limited number of vehicles (HVRP), **Taillard (1999)** presented an approach based on Tabu Search and Column Generation. **Gendreau et al. (1999)** proposed a Tabu Search algorithm, with different variable costs for each type of vehicle, to solve both the FSM-F and the FSM-D problems. The FSM-F problem was solved by **Ochi et al. (1998a, 1998b)** and **Osman and Salhi (1996)** using a Scatter Search approach and a Tabu Search algorithm, respectively. **Lima et al. (2004)** and **Liu et al. (2009)** proposed a Memetic Algorithm and a Genetic Algorithm, respectively, to solve the FSM-FD. **Tarantilis et al. (2003)** solved the HVRP-D by means of a list-based threshold accepting algorithm, in which a worse solution is only accepted if it is within a given threshold. **Tarantilis et al. (2004)** also introduced the Backtracking Adaptive Threshold Accepting algorithm to solve the HVRP-D. Finally, **Tarantilis et al. (2008)** proposed a Guided Tabu Search, to solve the heterogeneous fixed fleet vehicle routing problem (HFFVRP). Regarding exact approaches, **Baldacci and Mingozzi (2009)** and **Pessoa et al. (2009)** employed a Set-Partitioning Based algorithm and Branch-Cut-and-Price (BCP) algorithm, respectively. The Asymmetric and

Heterogeneous Vehicle Routing Problem (AHVRP), is another variant of the HVRP in which asymmetric distance-based costs are considered. **Herrero et al. (2014)** proposed a hybrid methodology for solving the AHVRP. Their approach combines a randomized version of a well-known savings heuristic with several local searches specifically adapted to deal with the asymmetric nature of distance-based costs.

4 Overview of our approach

As stated in the Introduction, our approach integrates, inside an ILS framework (**Lourenço et al., 2003**), biased-randomized versions of classical routing and packing heuristics. ILS is a conceptually simple yet powerful metaheuristic that has proven to be very efficient in solving a diversity of combinatorial optimization problems. The underlying idea behind ILS is to narrow the search for candidate local optimal solutions returned by some embedded algorithm. In order to do so, an ILS framework iteratively combines a perturbation stage with a local search stage (**Figure 2**). The perturbation stage aims at diversifying the search for a better solution inside the solution space, while the local search aims at intensifying the search for a better solution inside a local vicinity of the current solution. Usually, an acceptance criterion is also included to help the algorithm to escape from local minimum solutions. **Burke et al. (2010)** showed that ILS obtains the best average performance among a set of selected metaheuristic approaches in three classical combinatorial optimization problems: bin packing, permutation flow shop, and personnel scheduling. The authors also emphasize two main factors for its success: (i) an excellent balance between exploration and exploitation by “systematically combining a perturbation followed by local search”, and (ii) the reduced number of parameters required.

Biased-randomization of heuristics refers to the use of skewed probability distributions to transform fast deterministic heuristics into probabilistic algorithms **Juan et al. (2013a)**. In our case, we apply this technique over different routing and packing heuristics. For the routing part, we use a biased-randomized version of the popular saving heuristic (**Clarke and Wright 1964**). For the packing part, which is integrated inside the routing one for providing an integrated routing-and-packing optimization framework, we employ biased-randomized versions of two well-known packing algorithms: the Best-Fit packing (**Burke et al. 2004**) and the Touching Perimeter (**Lodi et al. 1999**). Moreover, our approach is also enriched with splitting and memory-based strategies as proposed in **Juan et al. (2011)**. The splitting technique is used during the perturbation stage to extract a subset of routes from a given feasible solution and then focus the improvement efforts on the associated sub-problem –notice that by improving the routing-and-packing plan for an independent subpart of a given solution, the entire solution will be also improved. Furthermore, the memory cache technique allows the algorithm to ‘remember’ good ways to cover a given set of customers as well as feasible packing configurations found in the previous iterations. Details on all these components of the algorithm are given in the next section.

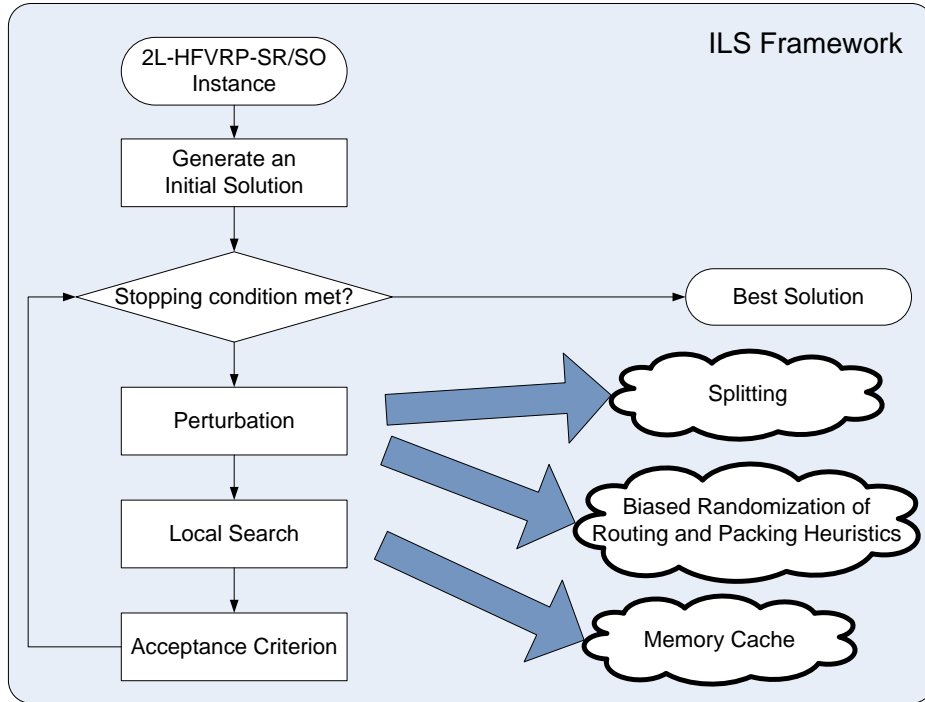


Figure 2: General scheme of our approach.

One important advantage of the proposed algorithm is its relative simplicity and, in particular, the fact that it employs just three parameters, as discussed in the next section. This reduces to a reasonable level the parameter fine-tuning process, which is often non-trivial and time-consuming. Finally, the approach proposed in this paper can be easily parallelizable: just by changing the seed for the pseudo-random number generator, different instances of the algorithm can be run in parallel using different threads, cores, or computers, thus providing high-quality results for complex combinatorial optimization problems in just a few seconds, as it has been already shown for similar frameworks and related vehicle routing problems (Juan et al. 2013b).

Finally, it is worthy to highlight at this point the main differences between the algorithm presented here and other previous algorithms we have developed in the past for solving simpler variants of the 2L-VRP:

- a) While our previous algorithms for simpler versions of the 2L-VRP followed a Multi-Start scheme, this algorithm is more evolved in the sense that it follows an Iterated Local Search (ILS) scheme (including the use of a base solution and an acceptance criterion). In general, ILS approaches –when properly set– tend to be more efficient –both in terms of solutions’ quality and in terms of computing times– than Multi-Start ones. In fact, a Multi-Start approach can be seen as a particular (extreme) case of an ILS.
- b) More importantly, the problem considered in this paper implies dealing with the sequencing issue during the distribution process. This additional constraint requires from specific strategies and procedures. In particular, both the Best-Fit packing heuristic (Burke et al. 2004) and the Touching Perimeter algorithm (Lodi et al. 1999) have been modified throughout a biased-randomized process so that they can efficiently deal with the sequencing constraint of the problem. Also, the new algorithm improves the packing procedure, including a memory cache technique for storing feasible packing solutions.

5 Low-level details of our approach

This section provides the pseudo-code details of the proposed algorithm. These details allow other authors and end users to quickly implement our algorithm in order to: (a) reproduce the experiments we have run and compare our approach against other approaches; and (b) use our approach to solve real-life applications of the 2L-HFVRP-SR. Thus, **Figure 3** shows the pseudo-code associated with the algorithm’s main procedure. Once the instance inputs (customers’ demands of items, available vehicles, and items’ sizes and weights) have been loaded into the program, an initial base solution is generated throughout the Pack-And-Route procedure (the generation process is explained later in detail). This initial base solution is also set as the best solution so far (**lines 1 and 2**). Then, the algorithm starts an ILS-like procedure (**lines 4 to 21**) that aims at improving the best solution by combining a destruction-construction (perturbation) stage of the base solution with a local search stage (**lines 5 to 9**). The perturbation and local search stages are based on splitting and cache techniques proposed in **Juan et al. (2011)** for the Capacitated Vehicle Routing Problem. In the perturbation stage, a set of adjacent routes is extracted from the base solution using a random selection process (destruction phase). This set of routes –together with its vehicle configuration and packing– constitutes a sub-solution for a 2L-HVRP-SR problem of smaller size than the original one (**line 6**) –and, therefore, easier to be efficiently solved.

```

Procedure ILS-BR(inputs,  $\alpha$ , maxPackIter,  $\beta$ )
01 baseSol  $\leftarrow$  packAndRoute(inputs,  $\alpha$ ,  $\beta$ ) % biased randomization
02 bestSol  $\leftarrow$  baseSol
03 delta  $\leftarrow$  0 % auxiliary variable
04 while {stopping condition not met} do
05   subSol  $\leftarrow$  extractRoutesAtRandom(baseSol) % destruction phase
06   subInputs  $\leftarrow$  getSubProblem(subSol)
07   newSubSol  $\leftarrow$  packAndRoute(subInputs,  $\alpha$ , maxPackIter,  $\beta$ ) % reconstruction phase
08   newSubSol  $\leftarrow$  applyCache(newSubSol) % fast local search
09   newSol  $\leftarrow$  merge(baseSol, newSubSol)
10   delta  $\leftarrow$  cost(newSol) - cost(baseSol)
11   if {delta < 0} then % base improvement
12     credit  $\leftarrow$  -delta
13     baseSol  $\leftarrow$  newSol
14     if {cost(newSol) < cost(bestSol)} then
15       bestSol  $\leftarrow$  newSol
16     end if
17   else if {0 < delta < credit} then % acceptance criterion
18     credit  $\leftarrow$  0
19     baseSol  $\leftarrow$  newSol
20   end if
21 end while
22 return bestSol
end procedure

```

Figure 3: Pseudo-code of our ILS-BR main procedure.

Next, the sub-problem is solved using the Pack-and-Route procedure (**line 7**), and the resulting sub-solution is merged with the non-extracted routes (**line 9**) to generate a new solution for the original problem (reconstruction phase). In fact, before this merging process occurs, a fast local search based on a cache of already computed routes is completed (**line 8**). Basically, for each route in the sub-solution, a hash map data structure containing the best found-so-far way to visit them is checked and, if appropriate, the route is updated. After the local search, the cost of the new solution is compared against the cost of the base solution (**line 10**). Whenever

the former is lower than the latter, the base solution is updated (**line 13**). A similar update process is made with our best-found solution if appropriated (**line 15**). Under some circumstances, the base solution will be updated with the new solution even if the latter shows a higher cost (**line 19**). This degradation of the base solution is performed occasionally in order to diminish the probability of the algorithm getting trapped in a local minimum. In our case, we use the following Demon-based criterion for accepting a downgrade of the base solution: (a) the size of the downgrade has to be inferior to the size of the last upgrade (represented in the pseudo-code by the credit variable); and (b) no two consecutive downgrades can be done (this is attained by resetting the credit to 0 anytime a downgrade is applied).

Figure 4 shows the pseudo-code for the Pack-and-Route procedure. First, a dummy solution is generated as described in the popular savings heuristic by **Clark and Wright (1964)**, i.e., a round-trip route from the depot to each customer is created (**line 1**). This initial solution employs the smallest possible vehicle on every route. Then, following the savings heuristic, the procedure computes the savings associated with each edge (**line 2**). These edges are initially sorted from highest to lowest savings and, afterwards, re-sorted according to a biased-randomization process –so that the higher the savings associated with one edge, the higher is the probability that the edge gets ranked at the top of the list (**line 3**).

```

Procedure packAndRoute(inputs,  $\alpha$ , maxPackIter,  $\beta$ )
01 sol  $\leftarrow$  genDummySol(inputs) % dummy CWS solution using smallest vehicles
02 savingsList  $\leftarrow$  buildSavingsList(inputs)
03 savingsList  $\leftarrow$  biasedRandSort(savingsList,  $\alpha$ ) % biased-rand of savings list
04 while {savingsList is not empty} do
05   nextEdge  $\leftarrow$  extractNextEdge(savingsList)
06   iNode  $\leftarrow$  getOriginNode(nextEdge)
07   jNode  $\leftarrow$  getDestinyNode(nextEdge)
08   iRoute  $\leftarrow$  getRoute(iNode, sol)
09   jRoute  $\leftarrow$  getRoute(jNode, sol)
10   vehType  $\leftarrow$  selectSmallestWeightCapableVehType(iRoute, jRoute)
11   isPackingFeasible  $\leftarrow$  false
12   isMergeAnImprovement  $\leftarrow$  false
13   while {veh in vehType has not been checked} do
14     isMergeAnImprovement  $\leftarrow$  checkIfMergeImprovesSol(iRoute, jRoute, veh)
15     if {isMergeAnImprovement is true} do
16       isPackingFeasible  $\leftarrow$  checkPacking(iRoute, jRoute, vehType, maxPackIter,  $\beta$ )
17       if {isPackingFeasible is false} then
18         if {vehType is not the largest type of vehicle} do
19           vehType  $\leftarrow$  selectNextLargerTypeOfVehicle(vehType)
20         else
21           exit while % refers to the nested one
22         end if
23       end if
24     end if
25   end while
26   if {isMergeAnImprovement} and {isPackingFeasible} then
27     newRoute  $\leftarrow$  merge(iRoute, jRoute, veh)
28     sol  $\leftarrow$  update(iRoute, jRoute, newRoute)
29   end if
30 end while
31 return sol
end procedure

```

Figure 4: Pseudo-code of the Pack-and-Route procedure.

As discussed in detail in [Juan et al \(2010, 2013a\)](#), this biased randomization of the savings list allows edges to be selected in a different order each time the procedure is called while, at the same time, the logic behind the savings heuristic is maintained. In our case, a skewed Geometric distribution is employed to induce this biased randomization behavior. The Geometric distribution uses one single parameter, α , which is relatively easy to set since $0 < \alpha < 1$. After completing some preliminary tests with different values for α and analyzing the corresponding outcomes, we decided to set $\alpha = 0.3$ in our computational experiments. At this point, the dummy solution is used as the initial solution in an iterative, route-merging, constructive process ([lines 4 to 30](#)). At each iteration, the edge at the top of the biased-randomized list is extracted, and the two routes connected by this edge are merged using the smallest possible vehicle if, and only if, the following conditions hold: (a) there is a vehicle available with enough capacity, in terms of weight, to carry the items coming from both merging routes; (b) the new merged route would reduce total costs –including both distance-based costs as well as costs associated with the type of vehicle being employed; and (c) the items from both routes can be conveniently loaded in the selected vehicle, i.e., they can be packed without overlapping and keeping the sequential order defined by the merged route. Notice that the checking of the packing feasibility might become a non-trivial and time-consuming process, which is discussed next. [Figure 5](#) shows the pseudo-code of the procedure that completes a fast checking of the packing feasibility. This procedure tries to quickly find a feasible solution that satisfies the weight capacity constraint, the packing-without-overlapping constraint, and the sequence-loading constraint.

```

Procedure checkPacking(iRoute, jRoute, vehType, maxPackIter,  $\beta$ )
01  isPackingFeasible  $\leftarrow$  false
02  itemList  $\leftarrow$  getItems(iRoute, jRoute)
03  custOrder  $\leftarrow$  getCustomerOrder(iRoute, jRoute)
04  packingSol  $\leftarrow$  emptySol
05  if {items total area  $\leq$  vehicle loading surface} then
06    packingSol  $\leftarrow$  checkPackingCache(custOrder, vehType) % checks memory cache
07    if {packingSol is not emptySol} then
08      isPackingFeasible  $\leftarrow$  true % feasible packing already in memory
09    else % try the biased-randomized version of the Touching Perimeter heuristic
10      packingSol  $\leftarrow$  biasedRandTP(itemList, custOrder, vehType, maxPackIter,  $\beta$ )
11      if {requestedLength(packingSol)  $\leq$  length(vehType)} then
12        update packing cache with packingSol
13        isPackingFeasible  $\leftarrow$  true
14      else % try the biased-randomized version of the Best Fit heuristic
15        packingSol  $\leftarrow$  biasedRandBF(itemList, custOrder, vehType, maxPackIter,  $\beta$ )
16        if {requestedLength(packingSol)  $\leq$  length(vehType)} then
17          update packing cache with packingSol
18          isPackingFeasible  $\leftarrow$  true
19        end if
20      end if
21    end if
22  end if
23  return isPackingFeasible
end procedure

```

Figure 5: Pseudo-code of the Check Packing Feasibility procedure.

Firstly, this procedure makes use of a cache memory (a hash map data structure) in order to quickly check whether a feasible packing solution was found before for the same configuration ([line 6](#)). If this is not the case, then the procedure employs biased-randomized versions of two fast and well-known packing heuristics ([lines 10 and 15](#)): the Touching Perimeter ([Lodi et al. 1999](#)) and the Best-Fit ([Burke et al. 2004](#)). Again, by employing

the biased-randomization technique described in [Juan et al. \(2011\)](#), these deterministic heuristics are transformed into probabilistic algorithms (*biasedRandTP* and *biasedRandBF*, respectively) that are likely to provide different outcomes each time they are run. These outcomes are then used to feed the memory cache, which keeps the best packing solution found so far for a given vehicle-items-order configuration. Basically, this randomization process aims at introducing some non-uniform randomness into the constructive part of the heuristic by employing a skewed probability distribution. In a similar way as we did for the routing part, a Geometric distribution of parameter β is used to induce the biased-random behavior in the packing heuristics. Again, we completed some preliminary tests with different values of β and, after analyzing the outcomes, we decided to randomly select a new β value, in the interval (0.06, 0.24), each time a packing heuristic is called. Finally, a maximum number of iterations is also included as a parameter. This parameter allows reducing the computation time employed in running the heuristics. In our experiments, we set this parameter to be twice the number of items to be packed.

6 Numerical experiments

The algorithm described in this paper has been implemented as a Java application. A single core of an Intel® Core™ i7-2670QM at 2.2 GHz and 4GB RAM was used to perform all the tests, which were run directly on the Netbeans platform installed over a Windows 7 operating system. In order to test our algorithm, we used the benchmarks described in [Leung et al. \(2013\)](#), which are a generalization of the classical 2L-CVRP benchmarks used in [Iori et al. \(2007\)](#) and [Gendreau et al. \(2008\)](#). The generalization was made by adding information about capacity, loading surface, fixed and variable costs for each of the four types of vehicles considered. In particular, we selected the benchmarks associated with the sequential loading, which are composed of 5 classes of instances –differing in the number and size of their items–, each class including 36 instances. Using our algorithm, the instances were run under two different scenarios, depending on whether or not the rotation of items was allowed. For each scenario, and for each instance-class combination, we run five replications of our algorithm, each of them using a different randomization seed. Each replication was run for a maximum time of 1 minute, i.e., a maximum time of 5 minutes was allowed for each scenario-instance-class combination. For the two-dimensional sequential oriented (without items rotation) loading scenario, 2|SO|L, we compare our results against the Simulated Annealing with Heuristic Local Search (SA_HLS) proposed in [Leung et al. \(2013\)](#). The SA_HLS was coded in C++ and it was run on a computer with a Core 2 Duo at 2.2 GHz and 2 GB RAM under Windows 7. As far as we know, for the two-dimensional sequential non-oriented (with items rotation) loading scenario, 2|SR|L, there are no previous results published. For that reason, we decided to compare the results we obtained for both scenarios as a way of estimating the reduction in costs derived from allowing rotation of items –whenever this might be a valid assumption. [Tables 1-3](#) show the results obtained for the three aforementioned approaches: SA_HLS (oriented), our best solution for the oriented scenario (OBS O), and our best solution for the scenario with rotation (OBS R). The values associated with the SA_HLS were directly obtained from [Leung et al. \(2013\)](#). The values associated with our methodology correspond to the best solution found after five runs. The computation time in seconds required to obtain the best solution in each case have been also included whenever they were available. With regard to Class 1, since all its items are square-shaped, results for this class are the same regardless items rotation is or is not allowed.

Finally, we have also tested the efficiency of our approach for the two-dimensional unrestricted oriented loading version, $2/UO/L$. **Table 4** shows a comparison between our algorithm and the aforementioned SA_HLS. In this case, the sequential loading constraint is removed, i.e., re-arrangements of the items inside the vehicle at the customers' sites are allowed. Once again, we include our best solution of the five runs, the CPU time elapsed and the percentage of improvement over the SA_HLS for this non-sequential loading version.

Table 1. Comparison for the $2|Sx|L$ between the SA_HLS ($x = O$) and OBS (both $x = O$ and $x = R$) - Classes 1 and 2.

Instance	Class 1					Class 2						
	SA_HLS (1)	t (s) (1)	OBS Oriented (2)	t (s) (2)	Gap (1)-(2)	SA_HLS (1)	OBS Oriented (2)	t (s) (2)	OBS Rotation (3)	t (s) (3)	Gap (1)-(2)	Gap (1)-(3)
1	596.07	29.78	596.07	0.00	0.00%	604.26	604.26	0.00	602.88	0.42	0.00%	-0.23%
2	679.18	5.05	670.30	0.00	-1.31%	721.60	721.60	0.06	700.94	0.00	0.00%	-2.86%
3	745.51	12.39	745.51	0.01	0.00%	772.90	772.90	0.26	765.80	0.02	0.00%	-0.92%
4	694.33	6.15	694.33	1.88	0.00%	721.90	709.13	0.03	697.06	0.02	-1.77%	-3.44%
5	761.19	5.62	758.47	0.02	-0.36%	799.80	783.08	0.44	762.92	0.30	-2.09%	-4.61%
6	809.56	4.77	809.56	0.07	0.00%	828.80	828.80	0.10	824.60	0.04	0.00%	-0.51%
7	3,211.53	3.67	3,211.53	0.35	0.00%	6,534.35	6,324.92	2.63	5,952.09	10.82	-3.21%	-8.91%
8	3,184.45	3.21	3,184.45	0.17	0.00%	5,097.06	5,025.73	1.14	4,804.14	0.64	-1.40%	-5.75%
9	1,029.95	7.39	1,029.95	0.01	0.00%	1,034.94	1,034.94	0.00	1,029.95	0.03	0.00%	-0.48%
10	5,149.51	6.97	4,932.34	0.87	-4.22%	8,342.56	7,990.31	8.93	7,439.25	32.00	-4.22%	-10.83%
11	5,119.40	6.97	4,932.34	0.88	-3.65%	8,410.74	8,182.93	13.33	8,007.51	58.36	-2.71%	-4.79%
12	1,658.56	35.80	1,655.73	0.31	-0.17%	1,674.56	1,674.56	0.04	1,674.56	0.04	0.00%	0.00%
13	14,655.40	1.93	14,579.87	0.15	-0.52%	28,771.60	27,546.67	46.70	26,318.79	56.68	-4.26%	-8.53%
14	10,019.00	13.51	9,097.52	0.09	-9.20%	11,580.30	11,005.84	30.82	10,389.98	2.79	-4.96%	-10.28%
15	10,151.70	5.49	9,097.52	0.08	-10.38%	11,402.30	11,076.43	23.38	10,780.29	39.42	-2.86%	-5.46%
16	1,292.58	17.94	1,276.22	0.01	-1.27%	1,285.83	1,285.83	0.10	1,280.90	0.03	0.00%	-0.38%
17	1,770.83	38.88	1,764.32	0.03	-0.37%	1,756.50	1,766.50	0.45	1,766.50	3.99	0.57%	0.57%
18	3,140.55	13.54	3,169.76	0.95	0.93%	6,044.31	6,029.64	25.25	5,862.61	40.71	-0.24%	-3.01%
19	1,553.11	32.56	1,490.28	0.18	-4.05%	4,559.17	4,578.73	55.76	3,955.68	55.51	0.43%	-13.24%
20	1,956.97	56.00	1,776.44	1.28	-9.23%	6,303.87	6,007.52	56.24	5,354.01	45.63	-4.70%	-15.07%
21	2,567.18	75.00	2,470.47	51.80	-3.77%	9,169.29	9,061.72	58.23	6,858.61	58.21	-1.17%	-25.20%
22	2,605.90	76.39	2,470.47	51.52	-5.20%	9,535.65	9,386.85	38.75	7,166.96	30.91	-1.56%	-24.84%
23	2,643.84	93.99	2,470.47	51.73	-6.56%	9,632.10	9,537.27	18.56	7,984.90	59.50	-0.98%	-17.10%
24	2,555.41	63.98	2,470.47	49.51	-3.32%	5,067.40	4,872.06	29.94	4,584.58	43.23	-3.85%	-9.53%
25	2,972.59	129.04	2,728.40	6.91	-8.21%	13,562.00	13,042.68	54.08	10,679.78	51.81	-3.83%	-21.25%
26	4,049.64	88.09	3,477.26	1.44	-14.13%	12,165.60	11,831.64	26.68	10,595.48	51.00	-2.75%	-12.91%
27	3,561.58	159.20	2,728.40	6.58	-23.39%	5,939.16	5,694.27	51.57	5,373.07	44.75	-4.12%	-9.53%
28	6,858.35	125.60	4,065.70	46.85	-40.72%	24,914.80	23,099.34	46.97	19,433.58	59.00	-7.29%	-22.00%
29	9,695.00	139.73	9,105.07	45.26	-6.08%	22,573.00	22,243.22	59.24	20,379.10	50.75	-1.46%	-9.72%
30	5,663.33	242.15	3,946.79	42.46	-30.31%	17,459.40	16,496.79	57.87	14,900.24	57.93	-5.51%	-14.66%
31	8,054.90	325.51	5,370.31	59.79	-33.33%	22,393.60	21,550.63	59.87	18,282.14	57.56	-3.76%	-18.36%
32	8,408.61	379.53	5,370.34	59.99	-36.13%	22,134.40	21,945.56	57.12	18,935.52	54.72	-0.85%	-14.45%
33	8,555.58	368.50	5,377.08	48.82	-37.15%	22,166.20	21,829.14	50.00	18,846.63	56.40	-1.52%	-14.98%
34	5,536.63	323.50	4,263.78	44.12	-22.99%	15,497.00	15,474.71	57.00	13,606.24	56.09	-0.14%	-12.20%
35	4,444.59	324.64	3,911.17	59.44	-12.00%	9,023.99	8,785.93	54.47	8,340.24	57.42	-2.64%	-7.58%
36	3,669.89	555.13	2,847.02	55.05	-22.42%	4,646.15	4,530.31	56.80	4,290.80	59.42	-2.49%	-7.65%
Averages	4,167.29	104.93	3,570.71	19.13	-9.71%	9,253.53	8,981.46	28.97	8,034.12	33.23	-2.09%	-9.46%

Table 2. Comparison for the 2|Sx|L between the SA_HLS (x = O) and OBS (both x = O and x = R) - Classes 3 and 4.

Instance	Class 3								Class 4							
	SA_HLS (1)	OBS Oriented (2)	t (s) (2)	OBS Rotation (3)	t (s) (3)	Gap (1)-(2)	Gap (1)-(3)	SA_HLS (1)	OBS Oriented (2)	t (s) (2)	OBS Rotation (3)	t (s) (3)	Gap (1)-(2)	Gap (1)-(3)		
1	597.28	609.81	0.00	597.04	0.01	2.10%	-0.04%	614.99	614.14	0.14	596.07	0.02	-0.14%	-3.08%		
2	734.36	728.24	2.54	728.24	36.46	-0.83%	-0.83%	684.99	695.91	0.02	695.91	0.02	1.59%	1.59%		
3	791.54	779.47	4.06	777.60	0.02	-1.53%	-1.76%	765.74	765.74	0.03	754.87	0.03	0.00%	-1.42%		
4	698.96	698.96	0.01	697.06	24.38	0.00%	-0.27%	703.76	703.76	0.49	703.76	0.01	0.00%	0.00%		
5	838.34	772.53	4.93	772.53	0.79	-7.85%	-7.85%	805.20	789.89	51.13	768.74	0.03	-1.90%	-4.53%		
6	836.09	836.09	9.86	827.60	0.04	0.00%	-1.01%	849.56	837.00	8.51	836.22	5.26	-1.48%	-1.57%		
7	5,456.98	5,216.26	0.06	5,135.66	0.03	-4.41%	-5.89%	5,648.18	5,596.04	15.96	5,466.90	21.34	-0.92%	-3.21%		
8	6,840.48	6,680.41	25.32	6,400.11	3.04	-2.34%	-6.44%	6,346.72	6,346.72	7.91	6,290.15	2.81	0.00%	-0.89%		
9	1,072.82	1,045.32	0.85	1,029.95	0.09	-2.56%	-4.00%	1,052.67	1,052.67	0.60	1,052.62	5.05	0.00%	0.00%		
10	7,045.68	6,580.03	35.64	6,173.72	47.17	-6.61%	-12.38%	8,408.75	7,927.68	8.61	7,977.48	14.29	-5.72%	-5.13%		
11	9,175.84	8,491.06	28.97	8,289.87	18.68	-7.46%	-9.66%	9,947.59	9,523.92	30.18	8,820.13	24.34	-4.26%	-11.33%		
12	1,681.51	1,681.51	0.11	1,681.51	0.05	0.00%	0.00%	1,685.18	1,680.01	0.74	1,680.01	0.89	-0.31%	-0.31%		
13	26,212.20	25,614.96	9.06	24,598.85	58.61	-2.28%	-6.15%	28,382.00	27,645.11	36.95	26,103.64	58.75	-2.60%	-8.03%		
14	11,415.60	10,944.31	56.88	11,345.99	42.35	-4.13%	-0.61%	10,965.80	10,781.90	5.93	10,233.70	53.03	-1.68%	-6.68%		
15	11,644.60	11,316.73	5.52	10,753.20	43.35	-2.82%	-7.66%	12,358.10	11,956.69	43.03	11,449.40	25.39	-3.25%	-7.35%		
16	1,301.60	1,294.93	0.08	1,288.07	0.14	-0.51%	-1.04%	1,296.62	1,290.00	0.19	1,290.00	0.20	-0.51%	-0.51%		
17	1,787.67	1,766.50	3.06	1,766.50	2.25	-1.18%	-1.18%	1,767.66	1,766.50	1.78	1,766.50	1.42	-0.07%	-0.07%		
18	5,761.77	5,478.57	53.06	5,197.55	50.79	-4.92%	-9.79%	6,416.46	6,343.99	45.76	6,084.39	13.46	-1.13%	-5.18%		
19	4,405.53	4,339.54	55.31	3,907.65	58.13	-1.50%	-11.30%	4,924.24	4,909.05	18.57	4,641.27	33.83	-0.31%	-5.75%		
20	6,517.72	5,964.79	43.59	5,468.92	44.83	-8.48%	-16.09%	6,676.43	6,151.25	49.42	5,689.54	46.89	-7.87%	-14.78%		
21	10,271.00	9,837.16	54.47	8,228.34	55.71	-4.22%	-19.89%	8,605.04	8,380.50	39.29	7,840.41	41.13	-2.61%	-8.89%		
22	9,583.69	9,229.27	59.42	8,117.96	58.69	-3.70%	-15.29%	9,614.78	9,120.47	39.24	8,107.14	54.28	-5.14%	-15.68%		
23	9,086.49	9,007.05	54.79	7,555.01	59.59	-0.87%	-16.85%	9,019.58	8,867.02	42.90	8,230.75	36.97	-1.69%	-8.75%		
24	4,713.20	4,566.74	45.68	4,271.53	59.73	-3.11%	-9.37%	4,937.93	4,670.16	56.07	4,520.48	35.47	-5.42%	-8.45%		
25	12,041.90	11,169.26	54.88	10,049.97	55.15	-7.25%	-16.54%	12,488.90	12,560.32	58.73	11,558.30	55.52	0.57%	-7.45%		
26	12,895.50	11,882.81	59.67	10,700.47	35.98	-7.85%	-17.02%	14,120.20	13,251.60	59.49	11,725.40	58.82	-6.15%	-16.96%		
27	6,273.66	6,003.14	58.86	5,735.30	39.99	-4.31%	-8.58%	5,895.78	5,820.64	55.72	5,497.14	58.59	-1.27%	-6.76%		
28	24,376.40	23,902.85	51.87	20,023.42	116.42	-1.94%	-17.86%	24,796.20	23,753.12	56.51	21,579.76	59.76	-4.21%	-12.97%		
29	22,746.10	21,236.21	46.88	20,343.62	59.25	-6.64%	-10.56%	24,164.20	22,950.32	58.28	21,568.48	55.08	-5.02%	-10.74%		
30	18,107.10	16,939.47	58.88	15,222.37	56.14	-6.45%	-15.93%	17,974.40	17,937.61	54.80	16,450.52	58.07	-0.20%	-8.48%		
31	23,643.40	22,106.28	59.46	20,243.98	56.75	-6.50%	-14.38%	24,192.90	23,476.65	49.59	21,480.92	59.72	-2.96%	-11.21%		
32	21,951.30	20,513.42	59.76	17,921.40	56.57	-6.55%	-18.36%	22,939.40	22,633.35	55.36	20,308.82	58.45	-1.33%	-11.47%		
33	23,392.00	22,804.37	58.99	20,344.29	59.33	-2.51%	-13.03%	24,245.10	23,910.80	41.49	21,899.41	57.40	-1.38%	-9.67%		
34	15,778.40	15,358.08	58.78	14,185.86	57.20	-2.66%	-10.09%	16,022.90	16,198.56	58.80	15,326.26	56.27	1.10%	-4.35%		
35	9,815.20	9,541.24	51.91	8,876.94	44.29	-2.79%	-9.56%	10,042.10	9,747.07	55.40	9,226.53	56.35	-2.94%	-8.12%		
36	4,866.30	4,779.03	59.26	4,524.41	53.38	-1.79%	-7.03%	4,643.86	4,521.06	59.45	4,280.03	57.80	-2.64%	-7.83%		
Averages	9,287.73	8,881.01	34.24	8,160.62	37.65	-3.51%	-9.01%	9,555.66	9,310.48	32.42	8,680.60	32.41	-2.00%	-6.56%		

Table 3. Comparison for the 2|Sx|L between the SA_HLS (x = O) and OBS (both x = O and x = R) - Classes 5 and Avg 2-5.

Instance	Class 5								Averages Classes 2-5							
	SA_HLS (1)	OBS Oriented (2)	t (s) (2)	OBS Rotation (3)	t (s) (3)	Gap (1)-(2)	Gap (1)-(3)	SA_HLS (1)	t (s) (1)	OBS Oriented (2)	t (s) (2)	OBS Rotation (3)	t (s) (3)	Gap (1)-(2)	Gap (1)-(3)	
1	596.07	596.07	0.07	596.07	0.05	0.00%	0.00%	603.15	5.73	606.07	0.05	598.02	0.12	0.48%	-0.85%	
2	679.18	670.30	0.00	670.30	0.01	-1.31%	-1.31%	705.03	6.09	704.01	0.66	698.85	9.12	-0.14%	-0.88%	
3	757.06	754.87	0.01	754.87	0.01	-0.29%	-0.29%	771.81	10.35	768.24	1.09	763.29	0.02	-0.46%	-1.10%	
4	694.87	694.87	0.05	694.87	0.02	0.00%	0.00%	704.87	8.71	701.68	0.14	698.19	6.11	-0.45%	-0.95%	
5	766.89	758.47	5.18	758.47	4.59	-1.10%	-1.10%	802.56	9.35	775.99	15.42	765.67	1.43	-3.31%	-4.60%	
6	824.60	824.60	0.03	822.69	0.01	0.00%	-0.23%	834.76	9.92	831.62	4.62	827.78	1.34	-0.38%	-0.84%	
7	5,443.79	4,945.64	10.95	4,723.64	0.44	-9.15%	-13.23%	5,770.83	1.95	5,520.71	7.40	5,319.57	8.16	-4.33%	-7.82%	
8	4,247.88	4,238.24	0.74	3,864.97	0.27	-0.23%	-9.01%	5,633.04	4.77	5,572.78	8.78	5,339.85	1.69	-1.07%	-5.20%	
9	1,029.95	1,029.95	0.04	1,029.95	0.03	0.00%	0.00%	1,047.60	15.74	1,040.72	0.37	1,035.62	1.30	-0.66%	-1.14%	
10	7,125.94	6,477.68	16.06	6,065.00	17.46	-9.10%	-14.89%	7,730.73	5.87	7,243.93	17.31	6,913.86	27.73	-6.30%	-10.57%	
11	6,433.59	6,116.72	5.19	6,053.52	46.91	-4.93%	-5.91%	8,491.94	10.07	8,078.66	19.42	7,792.76	37.07	-4.87%	-8.23%	
12	1,685.18	1,686.77	0.04	1,677.52	0.07	0.09%	-0.45%	1,681.61	32.9	1,680.71	0.23	1,678.40	0.26	-0.05%	-0.19%	
13	23,679.90	22,993.08	59.46	22,808.22	57.83	-2.90%	-3.68%	26,761.43	6.71	25,949.96	38.04	24,957.37	57.97	-3.03%	-6.74%	
14	10,519.50	10,271.23	25.07	10,005.07	10.64	-2.36%	-4.89%	11,120.30	11.29	10,750.82	29.68	10,493.68	27.20	-3.32%	-5.63%	
15	12,260.20	11,869.99	30.85	11,527.11	33.98	-3.18%	-5.98%	11,916.30	24.53	11,554.96	25.69	11,127.50	35.53	-3.03%	-6.62%	
16	1,280.90	1,280.90	0.06	1,280.90	0.04	0.00%	0.00%	1,291.24	25.73	1,287.92	0.11	1,284.97	0.10	-0.26%	-0.49%	
17	1,790.14	1,768.57	0.29	1,768.57	46.27	-1.20%	-1.20%	1,775.49	43.41	1,767.02	1.40	1,767.02	13.48	-0.48%	-0.48%	
18	4,939.95	4,471.79	53.54	4,435.96	14.62	-9.48%	-10.20%	5,790.62	30.5	5,581.00	44.40	5,395.13	29.89	-3.62%	-6.83%	
19	3,324.10	3,274.67	28.89	3,284.60	21.03	-1.49%	-1.19%	4,303.26	51.44	4,275.50	39.63	3,947.30	42.12	-0.65%	-8.27%	
20	5,363.89	4,459.31	56.83	4,133.83	52.69	-16.86%	-22.93%	6,215.48	98.14	5,645.72	51.52	5,161.57	47.51	-9.17%	-16.96%	
21	5,932.13	5,875.58	23.30	5,437.85	51.98	-0.95%	-8.33%	8,494.37	124.15	8,288.74	43.82	7,091.30	51.76	-2.42%	-16.52%	
22	6,734.27	6,034.44	59.63	5,899.70	38.45	-10.39%	-12.39%	8,867.10	110.4	8,442.76	49.26	7,322.94	45.58	-4.79%	-17.41%	
23	6,439.03	6,570.36	59.31	5,504.20	51.19	2.04%	-14.52%	8,544.30	130.44	8,495.42	43.89	7,318.72	51.81	-0.57%	-14.34%	
24	4,137.78	3,984.83	46.43	3,805.04	51.84	-3.70%	-8.04%	4,714.08	104.98	4,523.45	44.53	4,295.41	47.57	-4.04%	-8.88%	
25	8,316.47	7,693.50	43.87	7,227.21	59.62	-7.49%	-13.10%	11,602.32	186.28	11,116.44	52.89	9,878.82	55.53	-4.19%	-14.85%	
26	10,339.80	9,200.51	57.71	8,185.74	59.93	-11.02%	-20.83%	12,380.28	153.21	11,541.64	50.89	10,301.77	51.43	-6.77%	-16.79%	
27	5,422.40	5,359.72	58.95	5,101.17	55.91	-1.16%	-5.92%	5,882.75	240.04	5,719.45	56.27	5,426.67	49.81	-2.78%	-7.75%	
28	20,255.00	17,545.98	59.94	16,097.78	53.71	-13.37%	-20.52%	23,585.60	393.72	22,075.32	53.82	19,283.63	72.22	-6.40%	-18.24%	
29	22,271.90	21,025.83	56.81	20,143.84	57.79	-5.59%	-9.55%	22,938.80	486.55	21,863.90	55.30	20,608.76	55.72	-4.69%	-10.16%	
30	12,417.90	11,120.03	55.19	10,388.77	52.41	-10.45%	-16.34%	16,489.70	536.78	15,623.48	56.69	14,240.48	56.14	-5.25%	-13.64%	
31	17,902.30	16,477.95	59.95	14,853.19	59.75	-7.96%	-17.03%	22,033.05	1,110.8	20,902.88	57.22	18,715.06	58.44	-5.13%	-15.06%	
32	16,905.60	14,294.70	58.80	13,644.69	58.30	-15.44%	-19.29%	20,982.68	1,040.06	19,846.76	57.76	17,702.61	57.01	-5.41%	-15.63%	
33	17,823.30	15,828.48	58.30	13,648.79	59.50	-11.19%	-23.42%	21,906.65	1,087.93	21,093.20	52.19	18,684.78	58.16	-3.71%	-14.71%	
34	12,721.70	11,589.58	57.58	10,480.51	58.21	-8.90%	-17.62%	15,005.00	1,680.67	14,655.23	58.04	13,399.72	56.94	-2.33%	-10.70%	
35	8,372.87	8,138.06	57.30	7,776.65	52.74	-2.80%	-7.12%	9,313.54	1,722.96	9,053.07	54.77	8,555.09	52.70	-2.80%	-8.14%	
36	4,112.85	4,052.25	57.70	3,883.48	51.98	-1.47%	-5.58%	4,567.29	2,035.08	4,470.66	58.30	4,244.68	55.65	-2.12%	-7.06%	
Averages	7,598.58	7,054.88	32.34	6,639.85	32.79	-4.81%	-8.78%	8,923.88	321.03	8,556.96	31.99	7,878.80	34.02	-3.01%	-8.45%	

7 Discussion of results

For the sequential case, an initial look at **Tables 1-3** allows us to conclude the following: (i) for the non-oriented scenario, our algorithm is able to improve most of the individual results –as well as the average results– provided by the SA_HLS algorithm, with individual average gaps ranging from -2.00% (Class 4) to -9.71% (Class 1), and a total average gap for Classes 2 to 5 of -3.01%; (ii) results can be further enhanced, with respect to the oriented version of our algorithm, by allowing rotation of items in Classes 2 to 5; and (iii) computational times employed by our approach are far inferior to those employed by the SA_HLS algorithm using a similar CPU.

Since some of the gaps in Table 1 are noticeable, after checking again our results to make sure they were right we contacted the authors of the paper **Leung et al. (2013)** in order to ask for their results. They kindly reply with their latest results they have obtained (not published yet). These new results are now much closer to ours and, therefore, they support the gaps obtained by our algorithm.

Figure 6 allows comparing, for each class (class number between parentheses), the total costs associated with the best-found solution provided by each algorithm: SA-HLS (oriented), our algorithm oriented (OBS O), and our algorithm with rotation (OBS R). Notice that: (i) Class 1 is the one showing the lowest costs; (ii) inside each class, the average cost associated with OBS O is always lower than the one generated with the SA_HLS; and (iii) in each class (except class 1 where items rotation has no impact), the cost associated with OBS R represents always an additional improvement over the costs associated with OBS O –i.e., in practical situations, the possibility of rotating items contributes to reduce costs.

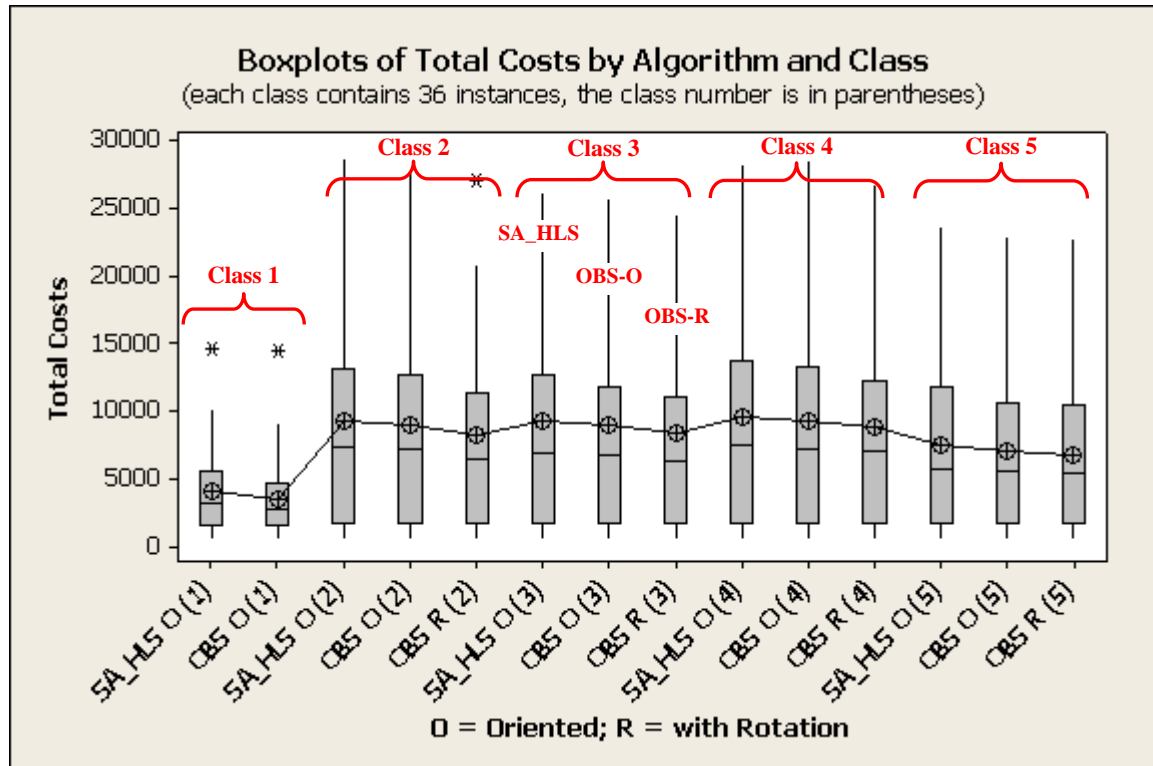


Figure 6. Comparison of Total Costs among SA_HLS Oriented, OBS Oriented, and OBS with Rotation.

Figure 7 allows comparing the gaps between results generated by pairs of algorithms and classes. In particular, notice the following: (i) when comparing the SA_HLS with OBS O, the greater gaps are those associated with classes 1 and 5; (ii) similarly, when comparing the SA_HLS with OBS R, the greater gaps are those associated with classes 1, 2, and 5 –of course, this comparison has to take into account that SA_HLS does not consider rotation of items, so the values provided by the SA_HLS are only used as references–; and (iii) comparing both the oriented and non-oriented scenarios of our algorithm for Classes 2 to 5, it follows that additional costs reductions (sometimes over 10% in size) can be attained by allowing the rotation of items during the loading stage, especially for classes 2 and 3.

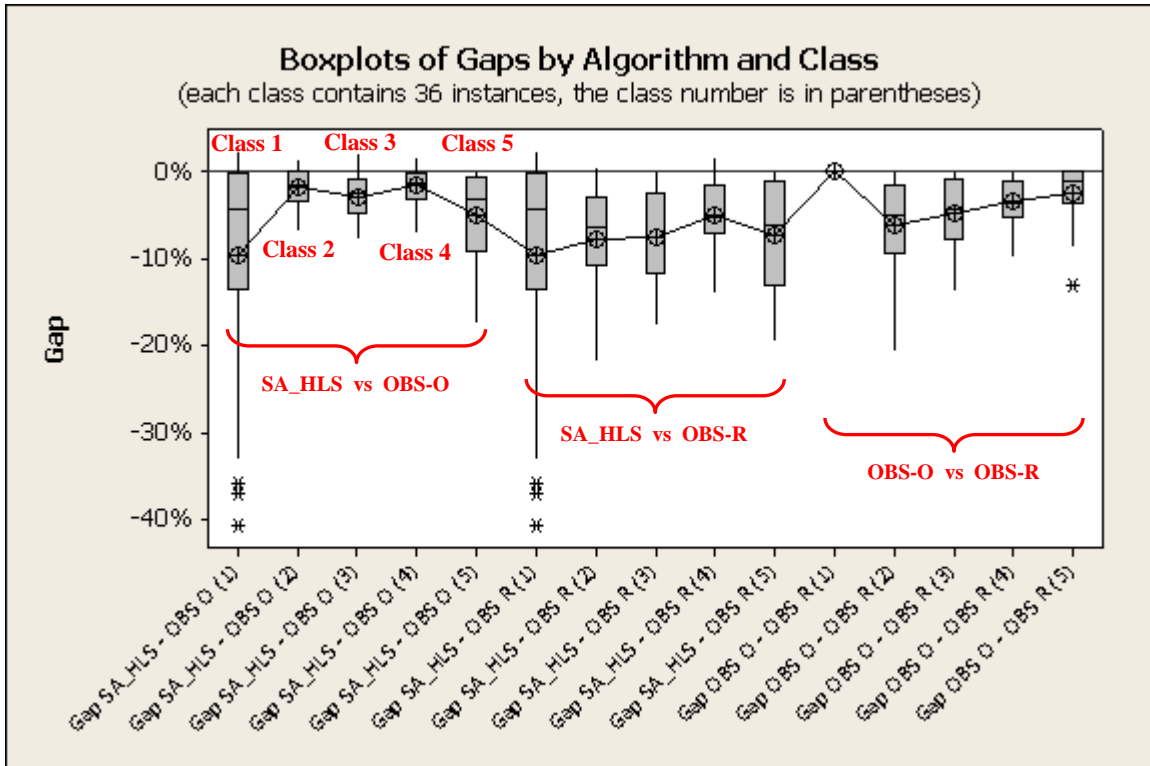


Figure 7. Comparison of Gaps among SA_HLS Oriented, OBS Oriented, and OBS with Rotation.

In order to show the complexity of the 2L-HFVRP-SR, **Figure 8** illustrates the sequential packing solution obtained for instance 10 in class 3.

Finally, regarding the unrestricted oriented loading case our approach also obtains high-quality solutions with low computational times. As we can see in **Table 4**, the improvements are even better than in the case of the sequential oriented loading version. Thus, the total average gap for Classes 2 to 5 between our approach and the SA_HLS is about 4.51%. As we would expect, the sequential loading constraint –in which the delivery order is taken into account– increases the average cost with respect to the unrestricted loading version of the problem – about 5.54% according to our results for the considered set of benchmarks.

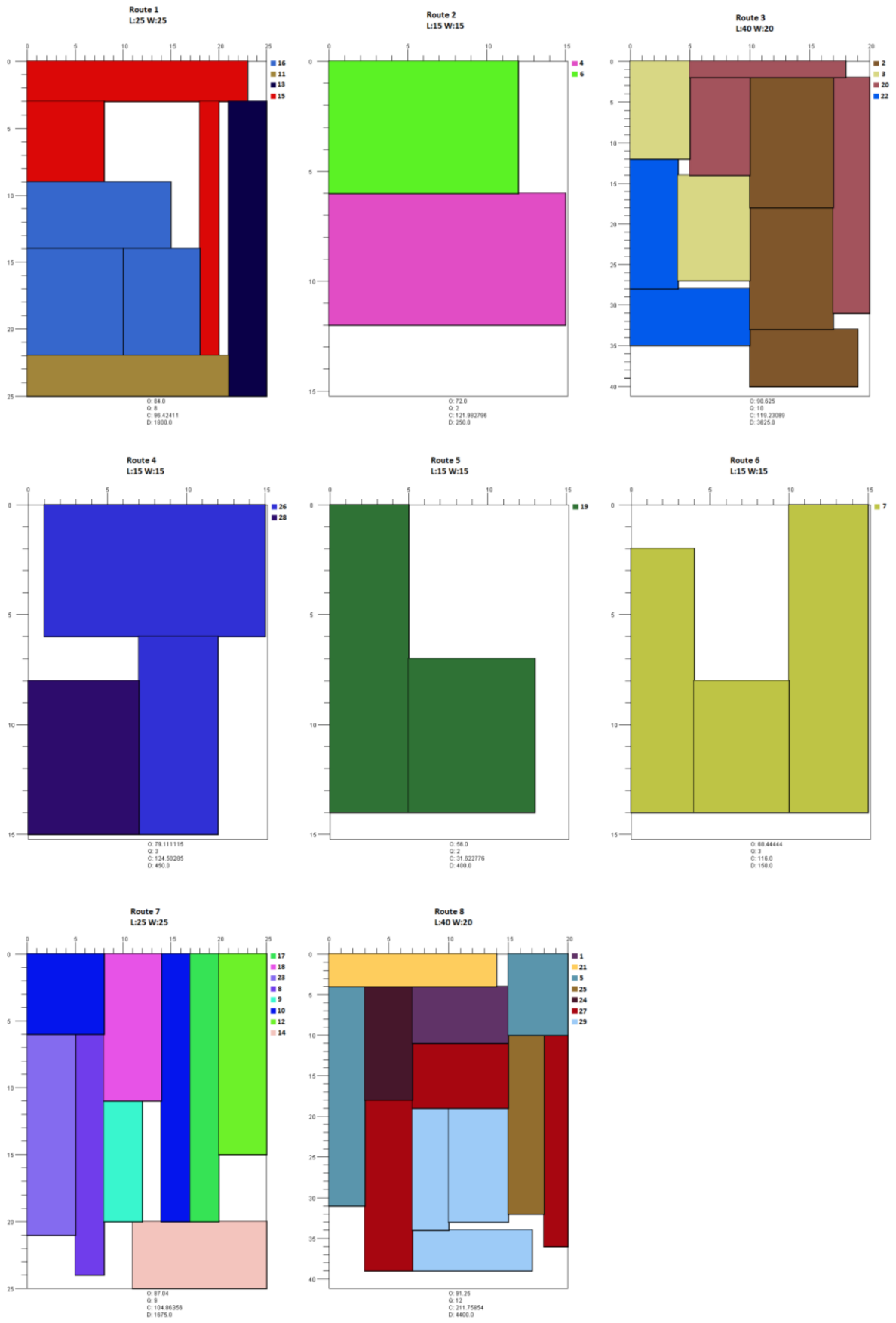


Figure 8. Sequential packing solution found for instance 10 in class 3.

8 Conclusions

This paper focuses on a rich and realistic version of the Two-Dimensional Vehicle Routing Problem (2L-VRP), which includes heterogeneity of vehicles, sequential loading, and items rotation. The version considered not only can be frequently encountered in real-life applications, but it also represents an important research challenge since it combines a heterogeneous vehicle routing problem with a highly-constrained packing problem. Although some recent papers have partially discussed the heterogeneous fleet, sequential loading, and items rotation versions of the 2L-VRP, to our knowledge this is the first work in the literature including all these conditions simultaneously. The paper presents a hybrid algorithm which combines biased-randomized versions of routing and packing heuristics inside an Iterated Local Search framework. Our approach considers both routing and packing costs simultaneously to better support the decision-making process. According to the results obtained, the algorithm proposed is able to provide state-of-the-art solutions to the richer version of the 2L-VRP and also to outperform other state-of-the-art approaches for the version without rotation.

Acknowledgments

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness (grant TRA2013-48180-C3-P), FEDER, and the Department of Universities, Research & Information Society of the Catalan Government (Grant 2014-CTP-00001).

References

- Baldacci, R., Battarra, M., & Vigo, D. (2008). Routing a heterogeneous fleet of vehicles. In *The vehicle routing problem: latest advances and new challenges* (pp. 3-27). Springer US.
- Baldacci, R., & Mingozzi, A. (2009). A unified exact method for solving different classes of vehicle routing problems. *Mathematical Programming*, 120(2), 347-380.
- Burke E. K., Kendall G., and Whitwell G. (2004). A new placement heuristic for the orthogonal stock-cutting problem. *Operations Research*, 52: 655–671.
- Burke, E., Curtois, T., Hyde M., Kendall G., Ochoa G., Petrovic S., Vázquez-Rodríguez J. A., Gendreau M. (2010). Iterated local search vs. hyper-heuristics: Towards general-purpose search algorithms. In *Evolutionary Computation (CEC), 2010 IEEE Congress on* (pp. 1-8). IEEE.
- Caceres J., Riera D., Juan, A., Arias, P., Guimarans, D. (2015). Rich Vehicle Routing Problem: Survey. *ACM Computing Surveys*.
- Clarke G. and Wright J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12: 568–81.
- Dominguez, O., Juan, A. A., Barrios, B., Faulin, J., & Agustin, A. (2014a). Using biased randomization for solving the two-dimensional loading vehicle routing problem with heterogeneous fleet. *Annals of Operations Research*, 1-22. doi: 10.1007/s10479-014-1551-4.

- Dominguez, O., Juan, A. A., & Faulin, J. (2014b). A biased-randomized algorithm for the two-dimensional vehicle routing problem with and without item rotations. *International Transactions in Operational Research*, 21: 375-398.
- Duhamel C., Lacomme P., Quilliot A., and Toussaint H. (2009). 2L-CVRP: A GRASP resolution scheme based on RCPSP. *International Conference on Computers & Industrial Engineering*, 2-9 July 2009. Troyes. Conference Publications, pp. 1094-1099. DOI: 10.1109/ICCIE.2009.5223772.
- Duhamel C., Lacomme P., Quilliot A., and Toussaint H. (2011). A multi-start evolutionary local search for the two-dimensional loading capacitated vehicle routing problem. *Computers & Operations Research*, 38(3): 617-640.
- Fuellerer G., Doerner K., Hartl R. & Iori M. (2009). Ant colony optimization for the two-dimensional loading vehicle routing problem. *Computers and Operations Research*, 36: 655-673.
- Gendreau, M., Laporte, G., Musaraganyi, C., & Taillard, É. D. (1999). A tabu search heuristic for the heterogeneous fleet vehicle routing problem. *Computers & Operations Research*, 26(12), 1153-1173.
- Gendreau M., Iori M., Laporte G. & Martello S (2008). A tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints. *Networks*, 51: 4-18.
- Golden, B., Assad, A., Levy, L., Gheysens, F., (1984). The fleet size and mix vehicle routing. *Computers & Operations Research* 11 (1), 49-66.
- Golden B., Raghavan S. & Wasil E. (eds.) (2008). *The Vehicle Routing Problem: Latest Advances and New Challenges*. Springer, New York.
- Herrero, R., Rodríguez, A., Cáceres-Cruz, J. & Juan, A. A. (2014). Solving vehicle routing problems with asymmetric costs and heterogeneous fleets. *International Journal of Advanced Operations Management*, 6(1), 58-80.
- Iori M and Martello S (2010). Routing problems with loading constraints. *TOP*, 18: 4-27.
- Iori M., Salazar J.J., Vigo D. (2007). An exact approach for the vehicle routing problem with two-dimensional loading constraints. *Transportation Science*, 41(2): 253-264.
- Juan A., Faulin J., Ferrer A., Lourenço H. & Barrios B. (2013a). MIRHA: multi-start biased randomization of heuristics with adaptive local search for solving non-smooth routing problems. *TOP*, 21 (1): 109-132.
- Juan A., Faulin J., Jorba, J., Cáceres J. & Marques J. (2013b). Using Parallel & Distributed Computing for Solving Real-time Vehicle Routing Problems with Stochastic Demands. *Annals of Operations Research*, 207: 43-65.
- Juan A., Faulin J., Jorba J., Riera D., Masip D. & Barrios B. (2011). On the use of Monte Carlo simulation, cache and splitting techniques to improve the Clarke and Wright saving heuristics. *Journal of the Operational Research Society*, 62(6): 1085-1097.
- Juan A., Faulin J., Ruiz R., Barrios B. & Caballe S. (2010). The SR-GCWS hybrid algorithm for solving the capacitated vehicle routing. *Applied Soft Computing*, 10: 215-224.
- Khebbache-Hadji, S., Prins, C., Yalaoui, A., & Reghioui, M. (2013). Heuristics and memetic algorithm for the two-dimensional loading capacitated vehicle routing problem with time windows. *Central European Journal of Operations Research*, 21(2), 307-336.

- Leung S. C., Zheng J., Zhang D. & Zhou X. (2010). Simulated Annealing for the Vehicle Routing Problem with Two-dimensional Loading Constraints. *Flexible Services and Manufacturing Journal*, 22, 1-2, 61-82.
- Leung, S. C., Zhou, X., Zhang, D., & Zheng, J. (2011). Extended guided tabu search and a new packing algorithm for the two-dimensional loading vehicle routing problem. *Computers & Operations Research*, 38(1), 205-215.
- Leung S. C., Zhang Z., Zhang D., Hua X. & M. K. Lim. (2013). A meta-heuristic algorithm for heterogeneous fleet vehicle routing problems with two-dimensional loading constraints. *European Journal of Operational Research*, 225:199-210.
- Lima, C. D. R., Goldberg, M. C., & Goldberg, E. F. G. (2004). A memetic algorithm for the heterogeneous fleet vehicle routing problem. *Electronic Notes in Discrete Mathematics*, 18, 171-176.
- Liu, S., Huang, W., & Ma, H. (2009). An effective genetic algorithm for the fleet size and mix vehicle routing problems. *Transportation Research Part E: Logistics and Transportation Review*, 45(3), 434-445.
- Lodi, A., Martello S. & Vigo, D. (1999). Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems. *INFORMS Journal on Computing*, 11:345–357.
- Lourenço, H. R., Martin O. C. & Stützle T. (2003). *Iterated local search*. In: Glover F and Kochenberger G (eds). *Handbook of Metaheuristics*. Kluwer Academic Publishers: Norwell, MA, pp.321–353.
- Malapert, A., Guéret, C., Jussien, N., Langevin, A., Rousseau, L.-M., (2008). Two-dimensional Pickup and Delivery Routing Problem with Loading Constraints. *1st Workshop on Bin Packing and Placement Constraints (BPPC'08)*, Paris.
- Ochi, L. S., Vianna, D. S., Drummond, L. M., & Victor, A. O. (1998a). *An evolutionary hybrid metaheuristic for solving the vehicle routing problem with heterogeneous fleet*. In *Genetic Programming* (pp. 187-195). Springer Berlin Heidelberg.
- Ochi, L. S., Vianna, D. S., Drummond, L. M., & Victor, A. O. (1998b). *A parallel evolutionary algorithm for the vehicle routing problem with heterogeneous fleet*. In *Parallel and Distributed Processing* (pp. 216-224). Springer Berlin Heidelberg.
- Osman I. H. & Salhi S. (1996). Local search strategies for the vehicle fleet mix problem. In: Rayward- Smith V J, Osman I H, Reeves C R and Smith G D, (eds). *Modern Heuristic Search Methods*. Wiley. Chichester, pp 131-153
- Pessoa, A., Uchoa, E., & Poggi de Aragão, M. (2009). A robust branch-cut-and-price algorithm for the heterogeneous fleet vehicle routing problem. *Networks*, 54(4), 167-177.
- Prins C. (2009). “A GRASP×Evolutionary Local Search Hybrid for the Vehicle Routing Problem”, in: F.B. Pereira and J. Tavares (Ed.), *Bio-inspired Algorithms for the Vehicle Routing Problem*, *Studies in Computational Intelligence*, publisher Springer, Berlin, 161: 35–53.
- Riff, M. C., Bonnaire, X., & Neveu, B. (2009). A revision of recent approaches for two-dimensional strip-packing problems. *Engineering Applications of Artificial Intelligence*, 22(4), 823-827.
- Taillard, É. D. (1999). A heuristic column generation method for the heterogeneous fleet VRP. *RAIRO-Operations Research*, 33(01), 1-14.

- Tarantilis, C. D., Kiranoudis, C. T., & Vassiliadis, V. S. (2003). A list based threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. *Journal of the Operational Research Society*, 54(1), 65-71.
- Tarantilis, C. D., Kiranoudis, C. T., & Vassiliadis, V. S. (2004). A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. *European Journal of Operational Research*, 152(1), 148-158.
- Tarantilis, C. D., Zachariadis, E. E., & Kiranoudis, C. T. (2008). A guided tabu search for the heterogeneous vehicle routing problem. *Journal of the Operational Research Society*, 59(12), 1659-1673.
- Toth P. & Vigo D. (2002). *The Vehicle Routing Problem*. SIAM Publishers: Philadelphia.
- Wang F., Tao Y. & Shi N. (2009). A survey on vehicle routing problem with loading constraints. *International Joint Conference on Computational Sciences and Optimization*, 2: 602–606.
- Zachariadis E. E., Tarantilis C. D. & Kiranoudis C.T. (2009). A guided tabu search for the vehicle routing problem with two-dimensional loading constraints. *European Journal of Operational Research*, 195: 729–743.
- Zachariadis, E. E., Tarantilis, C. D. & Kiranoudis, C.T. (2013). Integrated distribution and loading planning via a compact metaheuristic algorithm. *European Journal of Operational Research*, 228(1), 56-71.