Variable Neighborhood Search for the Elementary Shortest Path Problem with Loading Constraints

Telmo Pinto, Cláudio Alves(✉), and José Valério de Carvalho

Centro Algoritmi, Escola de Engenharia, Universidade do Minho,
4710-057 Braga, Portugal
{telmo,claudio,vc}@dps.uminho.pt

Abstract. In this paper, we address the elementary shortest path problem with 2-dimensional loading constraints. The aim is to find the path with the smallest cost on a graph where the nodes represent clients whose items may have different heights and widths. Beyond its practical relevance, this problem appears as a subproblem in vehicle routing problems with loading constraints where feasible routes have to be generated dynamically. To the best of our knowledge, there are no results reported in the literature related to this problem. Here, we explore a variable neighborhood search approach for this problem. The method relies on constructive heuristics to generate feasible paths, while improved incumbents are sought in different neighborhoods of a given solution through a variable neighborhood search procedure. The resulting variants of the algorithm were tested extensively on benchmark instances from the literature. The results are reported and discussed at the end of the paper.

Keywords: Shortest path problem · Loading constraints · Variable neighborhood search · Computational study

1 Introduction

The last years have seen an increased interest for rich routing problems that consider practical constraints arising in real settings. A good example is the vehicle routing problem with loading constraints for which many different contributions have been described recently [2,5,10]. The problem merges two well-known problems in the field of combinatorial optimization, namely the classical vehicle routing problem and the packing problem that results from the need of building feasible routes when the items involved are 2- or 3-dimensional objects. Clearly, the resulting problem is NP-hard since it combines two problems that are already NP-hard. A comprehensive survey related to this family of problems can be found in [4].

The elementary shortest path problem with loading constraints arises in particular in the context of vehicle routing problems where the items have at least

O. Gervasi et al. (Eds.): ICCSA 2015, Part II, LNCS 9156, pp. 474–489, 2015.
DOI: 10.1007/978-3-319-21407-8_34

two dimensions, and the capacity of the vehicles is a strong constraint. The problem occurs typically when feasible routes have to be generated dynamically as happens for example in column generation based approaches. Given this strong connection between the two problems, we will briefly review in the sequel the main aspects and contributions related to vehicle routing problems with loading constraints.

Some versions of the capacitated vehicle routing problem with loading constraints (L-CVRP) consider that the loading and unloading of the items must be done without moving the other items that are in the vehicle. Therefore, the position of the items inside the vehicles depends directly on the sequence by which the clients are visited. This constraint is known as a sequential or LIFO (*Last-In, First-Out*) constraint, and the corresponding routing problems are referred to as sequential L-CVRP. The problems that do not enforce this constraint are called unrestricted L-CVRP. Moreover, if the items can be rotated inside the vehicles, the resulting problem is denominated by rotated L-CVRP.

Due to the inherent complexity of the L-CVRP, the vast majority of the approaches described in the literature are based on heuristic approaches. The only exact method reported so far is due to Iori *et al.* [5], who described a branch-and-cut method to solve the problem with 2-dimensional items (2L-CVRP). Gendreau *et al.* [2] proposed the first heuristic method for the 2L-CVRP for both the unrestricted and sequential case. Their approach is based on tabu search, and it allowed the visit of infeasible solutions by penalizing them in the objective function. In [10], the authors describe a guided tabu search approach for the 2L-CVRP, which is enhanced with diversification procedures that penalize long arcs in the solution. The feasibility of the routes is checked by means of five different heuristics.

Some attempts in solving the 2L-CVRP exactly through column generation are reported in [7], while this approach is seen by Iori and Martello in [4] as worthwhile of investigation in order to determine effectively exact solutions for this problem. The Dantzig-Wolfe decomposition principle on which column generation approaches are based has been extensively applied to vehicle routing problems. The standard reformulation that results from this decomposition is a set partitioning problem which is solved typically by dynamic column generation. The related pricing subproblem is an elementary shortest path problem with additional resource constraints. This problem has received much attention in the literature [1,8,9]. Applying the Dantzig-Wolfe decomposition to the L-CVRP yields also a set partitioning problem which can be solved through column generation. The pricing subproblem remains an elementary shortest path problem, but now the resource constraints become 2- or 3-dimensional packing constraints which are much more difficult to handle than other standard constraints as the capacity constraints, for example.

To the best of our knowledge, the elementary shortest path problem with loading constraints has never been addressed in the literature. In this paper, we describe and analyse a solution approach based on variable neighborhood search for the problem with 2-dimensional items and sequential constraints. To generate

feasible routes, we use different constructive methods that handle the packing part of the problem through alternative strategies based on bottom-left and level packing placement rules. Local search is supported on several neighborhoods defined from both the routing and packing definition of the solutions. The combination of the different strategies described in this paper leads to different variants of the variable neighborhood search algorithm. The performance of these variants is evaluated and compared through extensive computational experiments on benchmark instances from the literature for the 2L-CVRP.

The paper is organized as follows. In Section 2, we define formally the elementary shortest path problem with loading constraints addressed in this paper, and we introduce the corresponding notation. In Section 3, we describe the constructive heuristics used to generate feasible solutions for the problem. The neighborhood structures and the details of our variable neighborhood search algorithm are described in Section 4. In Section 5, we report on the computational experiments performed to evaluate and compare the performance of our approach. Some final conclusions are drawn in Section 6.

2 The Elementary Shortest Path Problem with Loading Constraints

The elementary shortest path problem with 2-dimensional loading constraints (2L-ESPP) is defined on a graph $G = (V, E)$ with the set of nodes V representing the n clients of the problem plus the depot 0 from which the vehicle leaves initially and to which it should come back at the end of the visits, and E representing the set of edges of the graph. The travelling cost associated to the edges $(i, j) \in E$ will be denoted by c_{ij}. The loading area of the vehicle has a total width denoted by W and a maximum height of H units. Each client $i \in V \setminus \{0\}$ has b_i 2-dimensional items of width and height respectively equal to w_i and h_i. These dimensions are general, *i.e.* no particular constraints apply to the width and height of the items. The visit of a client implies that all his items are loaded on the vehicle. Hence, we assume that the load associated to any client $i \in V \setminus \{0\}$ fits in the vehicle. The 2L-ESPP consists in finding the minimum cost route for the vehicle that starts and ends at the depot 0 and that visits at most once the clients in V. Note that here we assume that there are negative costs for some edges of the graph, which happens typically when the problem is defined as a pricing subproblem of a column generation model for the 2L-CVRP.

In this paper, we address the case where the items of the clients have a fixed rotation, and where the sequential or LIFO constraint applies. The latter implies that, during the loading and unloading of the items of a given client, the items of all the other clients that are already in the vehicle cannot be moved. Furthermore, lateral movements of the items inside the vehicle are forbidden. Hence, the loading and unloading operations can only be done in a direction that is parallel to the left and right sides of the vehicle. Figure 2 illustrates the case of a feasible and an infeasible loading pattern for the instance of Example 1

according to this sequential constraint. In the example represented in this figure, the sequence of visits is $(0, 2, 1, 0)$. However, in the pattern (b) of Figure 2, one of the items of client 2 cannot be unloaded without moving first the items of client 1.

A solution for the 2L-ESPP is defined as a sequence of clients that starts and ends at the depot 0, together with a placement position for each item of the clients that are visited. Alternatively, the latter can be replaced by defining the sequence by which the items of each client should be loaded on the vehicle, and by defining the placement rule that is used. Let S denote the sequence of clients visited in a solution of the 2L-ESPP. We have

$$S = (s_1, s_2, \ldots, s_{|S|}),$$

with $s_1 = s_{|S|} = 0$, while $s_2, \ldots, s_{|S|-1}$ represent the clients visited by the corresponding route. The cost of the solution associated to S will be denoted by $z(S)$, *i.e.* $z(S) = \sum_{i=1}^{|S|-1} c_{s_i s_{i+1}}$. Moreover, let P define the order by which the items of the clients of S are placed in the vehicles, such that

$$P = (p_2, \ldots, p_{|S|-1}),$$

with $p_i = (p_i^1, p_i^2, \ldots, p_i^{b_{s_i}})$, $i = 2, \ldots, |S| - 1$, and p_i^j representing the index of the j^{th} item of client s_i to be placed in the vehicle.

The following example illustrates through a small instance the details related to the definition of the 2L-ESPP and its solutions.

Example 1. Consider the instance of the 2L-ESPP represented in Figure 1. The set of nodes V is composed by $n = 4$ clients and the depot 0, *i.e.* $V = \{0, 1, 2, 3, 4\}$. The costs c_{ij}, $(i, j) \in E$, are shown beside the edges of the graph. The items of each client are identified through the tuple (i, k), with i representing the client and k the index of the item $(k = 1, \ldots, b_i)$. A feasible solution for this instance is the following:

$$S = (0, 2, 1, 0) \qquad \text{and} \qquad \mathrm{P} = ((1, 3, 2), (1, 2)).$$

assuming that a standard bottom-left rule is used to place the items. The corresponding loading pattern is illustrated in Figure 2-(a). The cost of this solution is $z(S) = -5$. It is easy to see that all the items of the clients can be unloaded from the vehicle without lateral movements or moving the items of the other clients. Figure 2-(b) shows an alternative loading pattern that violates this sequential constraint.

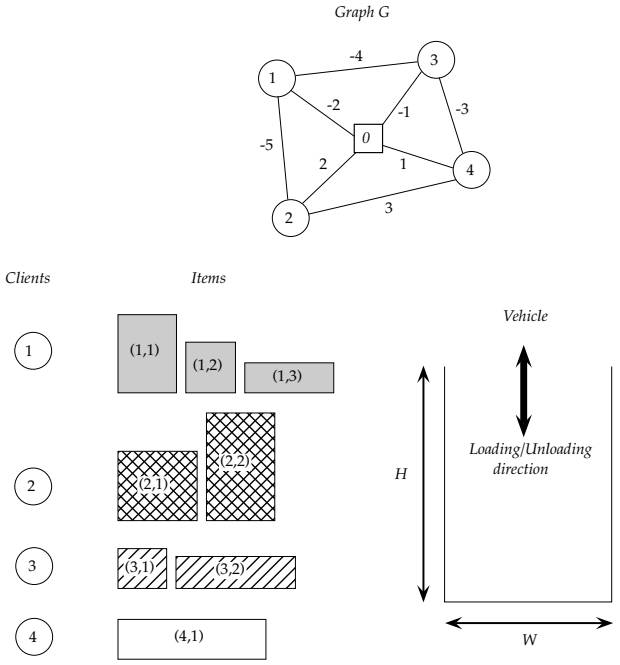

Fig. 1. Instance of the 2L-ESPP (Example 1)

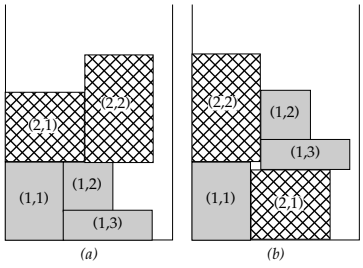

Fig. 2. Feasible (a) and infeasible loading pattern (b) (Example 1)

3 Building Feasible Solutions with Constructive Heuristics

To build an initial solution for the problem, we adopted a constructive approach in which the clients are added one by one to the route. The next client to be evaluated is inserted in the current route only if all his items can be loaded on the vehicle according to the constraints that apply to the loading patterns. The clients are evaluated following a nearest neighbor approach. Starting from the depot, two different strategies were considered to select the first client:

(FN) the first client to be evaluated is the one that is nearest to the depot 0 (*i.e.* $argmin\{c_{0i} : i = 1, \ldots, n\}$);
(FR) the first client to be evaluated is selected randomly.

The acronyms (FN) and (FR) used above will be used later to distinguish between the variants that we obtain by using one of these two strategies. After selecting the first client, the remaining ones are evaluated by non-decreasing order of the cost of the edge that connects them to the last client in the current route. The client that is evaluated is inserted in the route if all his items can be loaded on the vehicle. If no more clients can be inserted, the route is closed by connecting the last client to the depot.

The hardest part when building a solution for the 2L-ESPP is to find (if it exists) a feasible arrangement of the items on the vehicle such that there are no overlaps, all the items are put inside the boundaries of the vehicle without rotation, and the sequential constraint is satisfied. To address this issue, we considered three different strategies:

(BL) a standard bottom-left placement rule;
(RBL) a revised bottom-left procedure;
(LP) a level packing approach.

The strategy (BL) consists in placing the next item in the bottom and leftmost free position of the vehicle that ensures that all the loading constraints that apply are satisfied. Each time an item is placed on the vehicle, at most four orthogonal free spaces are generated identifying the different areas of the vehicle where the next items can be placed. In turn, after placing an item in a free space, this free space (and all the others that are intersected by the item) is removed, and replaced by an updated set of free spaces. Placing an item according to the strategy (BL) is equivalent to finding the free space whose width and height are equal to or larger than the size of the item, and whose bottom and leftmost position is the smallest among all the free spaces provided that it leads to a feasible placement. After finding this free space, the item is placed in its bottom and leftmost position. Figure 3 illustrates the concept of free space. Free space 1 and 2 (black dotted lines) are the free spaces respectively at the left and right of the item generated after its placement on the vehicle. Free spaces 3 and 4 (grey dashed lines) are the free spaces respectively above and below the item. The revised bottom-left procedure (RBL) consists in selecting the free space where

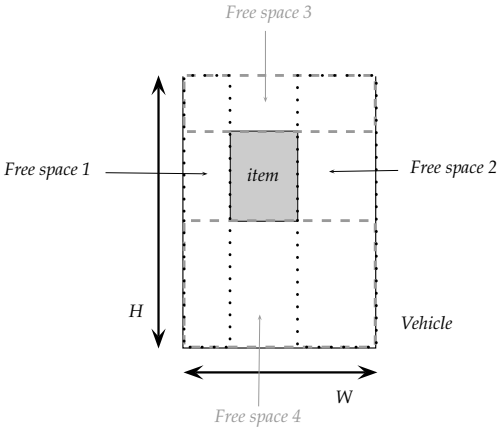

Fig. 3. Free spaces

the item produces the best fit, and then in placing the item in the bottom and leftmost position of this free space. The free space with the best fit is the one whose area is the nearest to the area of the item. The idea is to place the items in the areas of the vehicle where the packing generates the least possible waste, thus favouring the filling of holes. Again, only the free spaces whose bottom and leftmost position yields a feasible placement are considered for selection.

The last strategy (LP) that we explored consists in placing the items of the selected client in horizontal levels, and then in placing the levels on the vehicle one above another. The first item to be placed in a level determines the maximum height of the items that can be placed after it in that level. To select the level where the next item will be placed, we considered the following two strategies:

($LP.FF$) the next item is placed in the first open level where it fits;
($LP.BF$) the next item is placed in the open level where it best fits.

The level that was placed in the upmost position of the vehicle for the previous client is considered as an open level when placing the items of the next client. After placing all the items on the levels, the levels are placed on the vehicle so that the one with the largest remaining space is placed in the upmost position. One of the advantages of level packing approaches is that they ensure that the patterns satisfy the sequential constraint, thus avoiding the necessity of checking the placement positions before placing the items. The loading patterns resulting from this level packing approach are similar to the cutting patterns that arise in 2-dimensional guillotine cutting stock problems.

After choosing a client to insert in the route, his items are selected one by one, and placed in the vehicle according to the strategies described above. The next item to be placed is selected according to two different orderings based on the following criteria:

(OH) height of the items;
(OA) area of the items.

In both cases, the items are ordered in non-increasing order of their height and area, respectively.

4 Variable Neighborhood Search Algorithm

The strategies described in the previous section yield different variants of a constructive method for building feasible solutions for the 2L-ESPP. In order to improve the solutions generated by these algorithms, we developed a local search approach embedded into a variable neighborhood search (VNS) algorithm. The VNS metaheuristic was described first by Mladenovic and Hansen in [6], and since then, it has been applied with success to solve different combinatorial optimization problems [3]. The aim of VNS is to explore systematically different neighborhoods of the solutions to diversify the search and escape from local optima. Here, VNS is used to drive the search into 7 alternative neighborhoods of the solutions generated through the constructive algorithms arising from the combination of the different strategies described above. In the following section, we define the neighborhood structures that we used to support the local search procedures. The details of our VNS algorithm are given in Section 4.2.

4.1 Neighborhood Structures

The representation of the solutions of the 2L-ESPP defined in Section 2 relies on two main elements: the sequence by which the clients are visited in a given route, and the characterization of the loading pattern used to arrange the corresponding items in the vehicle so that all the loading constraints that apply (no overlaps, fixed orientation and the sequential constraint) are satisfied. To explore the search spaces defined through these two aspects of the solutions, we defined 7 neighborhood structures that can be divided into routing neighborhoods and packing neighborhoods. The definition of the neighborhood structures relies on the constructive algorithms defined above, and covers a broad range of possible movements. Let CH denote the constructive heuristic used to build the initial solution for the instance, and which is obtained by combining the strategies described in Section 3. In our implementation, we assumed that the constructive heuristic CH used to generate the initial solution is also the one that is used to define the neighbors of a given solution. When generating the neighbors, the difference is that part of the sequences of clients (and the corresponding sequence by which the items are placed in the vehicle) is fixed. The neighborhood structures are defined in the sequel.

Routing neighborhoods

NS_1 **Swapping two clients in the route**
Given a solution whose sequence of visited clients is S (as defined in Section 2), the neighbors of this solution consist in all the feasible solutions

obtained by swapping two clients in this route, by applying CH with the resulting sequence of clients (keeping the sequence of items for each client), and by adding more clients from the last one forward in the route using CH. Let S_c denote the sequence of clients in the route associated to the current solution, such that:

$$S_c = (s_1, s_2, s_3, \ldots, s_{|S_c|-1}, s_{|S_c|}),$$

One of the neighbors of this solution obtained by swapping s_2 and $s_{|S_c|-1}$ is the following:

$$S'_c = (s_1, s_{|S_c|-1}, s_3, \ldots, s_2, s_{|S_c|}),$$

provided that the items of the clients in S'_c can be put in the vehicle using CH, and no more clients can be added at the end of S'_c. Moreover, if P_c defines the sequences by which the items of the clients of S_c are placed in the vehicle (again as defined in Section 2), *i.e.*

$$P_c = (p_2, p_3, \ldots, p_{|S_c|-1}),$$

then the corresponding sequences of items associated to S'_c will be

$$P'_c = (p_{|S_c|-1}, p_3, \ldots, p_2).$$

NS_2 **Shifting a client in the route**
The neighbors of a solution whose sequence of clients is S are obtained by choosing a client and placing it in a different position of the sequence, by applying CH with the resulting sequence of clients (keeping the sequence of items for each client), and by adding more clients from the last one forward in the route using CH. The following solution is a neighbor of the solution S_c defined above obtained by selecting the client s_2 and by placing it in the third position of the sequence, *i.e.*

$$S'_c = (s_1, s_3, s_2 \ldots, s_{|S_c|-1}, s_{|S_c|}).$$

Again, in this case, we are assuming that no more clients can be added at the end of the route by applying CH.

NS_3 **Removing a client from the route**
The neighbors of a solution are obtained by removing a client from the route, by applying CH with the resulting sequence of clients in the same conditions as in the previous neighborhood structures, and by inserting clients at the end of the sequence (before the depot and if they fit in the vehicle) using again CH.

NS_4 **Removing a client and all its successors from the route**
The neighbors of a solution are obtained by removing all the clients from a given position of the sequence up to the end, by adding a selected client

at the end of the sequence and by placing his items using CH. The vehicle is filled by applying strictly the heuristic CH starting from the last client that was inserted.

NS_5 **Exchanging a client by another in the route**
The neighbors of a solution are obtained by exchanging a client by another that is not in the sequence, by placing the items of the clients in the resulting sequence using CH, and by adding other clients at the end of the sequence (before the depot) using again CH. Note that the sequence by which the items of the clients are placed in the vehicle remains unchanged for all the clients that were already in the route.

Packing neighborhoods

NS_6 **Swapping two items**
The neighbors of a solution are obtained by selecting a client in the route and by swapping two items in the sequence that defines the order by which his items are placed in the vehicle. Then, the heuristic CH is used to build the solution that corresponds to these sequences of clients and items (if possible), and to add other clients at the end of the sequence (before the depot) if they fit in the vehicle. Let S^c be the sequence of visited clients in the current solution, and let P^c denote the corresponding sequences by which the items are placed in the vehicle. As an example, let S^c and P^c be defined respectively as follows:

$$S_c = (s_1, s_2, s_3, s_4, s_5),$$

with $s_1 = s_5 = 0$, and

$$P_c = ((p_2^1, p_2^2), (p_3^1, p_3^2, p_3^3), (p_4^1)).$$

A possible neighbor S'_c of this solution is defined as follows:

$$S'_c = S_c \qquad \texttt{and} \qquad P'_c = ((p_2^1, p_2^2), (p_3^2, p_3^1, p_3^3), (p_4^1)).$$

It is obtained by swapping the first and second item of s_3 in P_c, provided that all the items can be placed according to P'_c in the vehicle by applying CH, and that no more clients can be added at the end of the sequence.

NS_7 **Shifting an item**
The neighbors of a solution are obtained by selecting a client in the route and by placing one of his items in a different position in the sequence that defines the order by which the items of this client were inserted in the vehicle. As in the previous neighborhood structure, the heuristic CH is used to build the solution associated to these sequences of clients and items. The same heuristic is used to add other clients at the end of the sequence and before the depot, if possible.

4.2 Variable Neighborhood Search

To explore the search spaces defined through the neighborhood structures described in the previous section, we developed a variable neighborhood search algorithm that applies local search on these 7 neighborhoods. The initial solution is generated by applying one of the constructive heuristics that result from the combination of the different strategies described in Section 3, namely $\{(FN), (FR)\}$, $\{(BL), (RBL), (LP)\}$, $\{(LP.FF), (LP.BF)\}$ (if (LP) has been selected), and $\{(OH), (OA)\}$. Then, the 7 neighborhoods are explored in cycle until a maximum computing time limit is reached. A solution is generated in a shaking phase from the current incumbent solution in the neighborhood that is being explored, and a local search procedure is applied right after in the same neighborhood to determine an improved solution. In our implementation, we resorted to a first improvement local search procedure that stops when it finds a solution that is better than the solution generated in the shaking phase, or if no better solution exists in this neighborhood. Note that all the solutions that are explored are necessarily feasible solutions for the problem. Our variable neighborhood search algorithm is described in Algorithm 1. The constructive heuristic is denoted by `findInitialSolution()`, while the shaking and local search procedures are represented respectively by `shaking`$((S, P), NS_k)$ and `firstImprovement`$((S', P'), NS_k)$, with (S, P) denoting the current incumbent solution, (S', P') the solution generated in the shaking phase, and NS_k the neighborhood that is being explored.

5 Computational Experiments

To evaluate and compare the performance of the different variants of our variable neighborhood search algorithm, we conducted an extensive set of computational experiments on 180 benchmark instances of the 2L-CVRP used by Iori *et al.* in [5] and by Gendreau *et al.* in [2]. Note that, in the former, the largest instances were not used due to their complexity. The number n of clients of these instances ranges from 15 up to 255, while the total number of items varies between 15 and 786. A complete description of the instances can be found in [2]. For our experiments, we multiplied all the costs (distances) associated to the edges by -1. The algorithms were coded in C++, and the tests were run on a PC with an i7 CPU with 2.9 GHz and 8 GB of RAM.

We tested the 16 variants of our algorithm resulting from the combination of the strategies described in Section 3, namely $\{(FN), (FR)\}$, $\{(BL), (RBL), (LP)\}$, $\{(LP.FF), (LP.BF)\}$ (if (LP) has been selected), and $\{(OH), (OA)\}$. The instances were divided in 22 groups according to the number of clients. The average results for each group are reported in Tables 1-3. In Table 1, we report on the results obtained with the standard bottom-left placement rule (BL) for all the possible combinations of strategies involving (FN), (FR), (OH) and (OA). Table 2 provides the results achieved with the revised bottom-left placement rule (RBL), while Table 3 gives the results when the level packing procedure (LP) is used with $(LP.FF)$. Although we tested the

Algorithm 1. Variable Neighborhood Search Algorithm

Input:
I: instance of the 2L-ESPP;
CH: constructive heuristic defined from the combination of the different strategies $\{(FN), (FR)\}$, $\{(BL), (RBL), (LP)\}$, $\{(LP.FF), (LP.BF)\}$ (if (LP) has been selected), and $\{(OH), (OA)\}$;
Set of neighborhood structures $NS = \{NS_1, NS_2, \ldots, NS_7\}$;
Limit t_{max} on the total computing time;
Output:
Feasible solution (S, P) of value $z(S)$;

(S, P):=`findInitialSolution()`;
repeat
  $k := 1$;
  while $k \leq 7$ **do**
    $(S', P') :=$ `shaking`$((S, P), NS_k)$;
    $(S'', P'') :=$ `firstImprovement`$((S', P'), NS_k)$;
    if $z(S'') \leq z(S)$ **then**
      $(S, P) := (S'', P'')$;
      $k := 1$;
    end
    else
      k:=k+1;
    end
  end
until *cpuTime()* $\leq t_{max}$;
return (S, P) ;

level packing procedure (LP) with the best-fit rule $(LP.BF)$, we do not report the results here due to the lack of space. This latter strategy led to results that are very near from those obtained with the approach based on $(LP.FF)$ for these instances. All the tests were run with a maximum computing time limit of 3 seconds. The purpose was to test and compare the ability of each strategy in finding efficiently good incumbents for the problem. The meaning of the columns in these tables is the following:

n: number of clients;
M: average number of items;
$inst$: number of instances in the corresponding group;
ord: criterion used to order the items of a client ((OH) or (OA));
z_{CH}: average value of the initial solution generated using the constructive heuristic resulting from the combination of the strategies described in Section 3;
$\%_{fill}^{CH}$: average percentage of space used in the vehicle by the initial solution generated using the constructive heuristic;
z_{VNS}: average value of the best solution obtained with the variable neighborhood search algorithm described in Algorithm 1;
$\%_{fill}^{VNS}$: average percentage of space used in the vehicle by the best solution obtained with the variable neighborhood search algorithm;

imp: percentage of improvement achieved with the variable neighborhood search algorithm, *i.e.* $imp = (z_{VNS} - z_{CH})/z_{CH}$.

Additionally, in Tables 1-3, the average results for all the instances are reported in the line *avg*.

Table 1. Computational results with (BL)

				(FN)					(FR)				
n	M	$inst$	ord	z_{CH}	$\%_{fill}^{CH}$	z_{VNS}	$\%_{fill}^{VNS}$	imp	z_{CH}	$\%_{fill}^{CH}$	z_{VNS}	$\%_{fill}^{VNS}$	imp
15	31,1	10	(OH)	-335,30	61,26	-349,30	65,07	4,18	-318,90	61,21	-341,30	63,87	7,02
20	39,5	10		-423,80	60,82	-483,00	67,33	13,97	-435,30	63,02	-475,70	66,56	9,28
21	39,4	10		-561,90	63,93	-621,70	69,91	10,64	-542,70	65,63	-594,00	66,81	9,45
22	39,4	10		-940,80	62,23	-1076,70	68,42	14,45	-966,50	64,32	-1049,40	68,26	8,58
25	56,0	5		-532,20	60,12	-561,80	66,56	5,56	-452,00	61,12	-550,60	67,96	21,81
29	57,8	10		-984,70	63,54	-1137,00	69,39	15,47	-969,90	62,10	-1149,50	65,53	18,52
30	63,8	5		-512,40	62,72	-561,00	69,22	9,48	-497,60	65,70	-556,20	69,92	11,78
32	62,5	15		-1718,87	61,84	-1831,33	69,02	6,54	-1619,07	60,55	-1858,00	69,52	14,76
35	74,4	5		-576,20	60,74	-648,80	68,46	12,60	-554,40	62,68	-639,80	63,40	15,40
40	79,2	5		-611,60	57,78	-754,20	71,58	23,32	-643,60	63,26	-766,80	69,46	19,14
44	86,2	5		-1250,40	58,76	-1432,00	70,46	14,52	-1262,80	63,70	-1431,00	70,26	13,32
50	105,2	5		-770,80	62,14	-853,60	70,30	10,74	-717,20	63,42	-826,20	71,18	15,20
71	146,0	5		-542,00	61,40	-607,00	70,20	11,99	-548,60	68,76	-602,20	72,96	9,77
75	150,3	20		-1068,45	64,47	-1215,80	70,75	13,79	-1038,85	63,16	-1201,70	70,69	15,68
100	204,3	15		-1399,80	65,99	-1561,00	72,73	11,52	-1376,00	65,90	-1554,40	72,47	12,97
120	245,6	5		-2552,20	70,72	-2659,00	73,26	4,18	-2483,40	68,16	-2711,20	73,06	9,17
134	271,4	5		-2545,80	70,42	-2805,00	73,52	10,18	-2545,80	69,58	-2748,20	76,90	7,95
150	294,4	5		-1880,40	70,14	-2026,40	74,20	7,76	-1838,20	65,48	-2025,80	71,90	10,21
199	399,6	15		-2308,40	67,25	-2490,47	74,90	7,89	-2225,07	68,77	-2442,20	72,68	9,76
240	484,8	5		-1133,00	64,64	-1247,80	74,98	10,13	-1138,20	73,66	-1228,40	76,54	7,92
252	504,4	5		-1439,60	65,14	-1521,00	77,68	5,65	-1416,60	62,96	-1516,80	77,80	7,07
255	509,0	5		-1022,00	67,26	-1124,80	77,54	10,06	-1030,80	71,20	-1092,00	77,42	5,94
			avg.	**-1141,39**	**63,79**	**-1253,12**	**71,16**	**10,67**	**-1119,16**	**65,20**	**-1243,70**	**70,69**	**11,85**
15	31,1	10	(OA)	-331,40	60,03	-351,40	62,74	6,04	-288,30	57,59	-348,00	65,21	20,71
20	39,5	10		-412,10	60,50	-477,10	66,77	15,77	-398,90	63,40	-468,90	66,70	17,55
21	39,4	10		-546,50	61,69	-612,10	67,42	12,00	-554,40	62,60	-619,30	69,18	11,71
22	39,4	10		-958,20	64,97	-1042,20	68,73	8,77	-914,90	60,91	-1052,10	67,20	15,00
25	56,0	5		-539,80	64,60	-576,20	67,60	6,74	-498,20	63,44	-581,40	70,00	16,70
29	57,8	10		-947,70	62,79	-1146,50	69,05	20,98	-980,80	62,76	-1145,10	67,06	16,75
30	63,8	5		-519,00	64,88	-541,60	66,46	4,35	-505,40	62,72	-542,20	67,98	7,28
32	62,5	15		-1718,93	60,99	-1852,87	67,44	7,79	-1617,80	61,50	-1891,47	69,35	16,92
35	74,4	5		-614,40	61,28	-651,00	66,64	5,96	-622,00	65,72	-666,80	69,14	7,20
40	79,2	5		-669,40	64,90	-727,60	70,74	8,69	-690,20	63,78	-740,20	69,06	7,24
44	86,2	5		-1250,40	58,76	-1468,00	68,66	17,40	-1226,40	64,44	-1411,40	67,84	15,08
50	105,2	5		-777,60	66,76	-826,60	66,28	6,30	-763,60	67,10	-865,40	70,86	13,33
71	146,0	5		-542,80	59,98	-587,60	72,30	8,25	-550,40	61,26	-614,60	71,64	11,66
75	150,3	20		-1098,30	67,02	-1211,10	73,04	10,27	-1032,45	63,61	-1207,85	70,57	16,99
100	204,3	15		-1411,00	67,97	-1575,00	72,86	11,62	-1414,87	70,09	-1573,73	72,59	11,23
120	245,6	5		-2548,60	71,56	-2706,60	73,10	6,20	-2389,20	69,76	-2680,60	72,44	12,20
134	271,4	5		-2562,40	71,24	-2724,00	71,46	6,31	-2365,40	71,04	-2710,40	72,84	14,59
150	294,4	5		-1884,80	70,00	-2003,80	74,18	6,31	-1788,00	66,22	-1965,20	75,50	9,91
199	399,6	15		-2286,27	69,96	-2448,40	76,00	7,09	-2252,60	70,74	-2477,67	74,36	9,99
240	484,8	5		-1157,00	71,94	-1227,00	77,50	6,05	-1153,20	73,22	-1249,60	76,54	8,36
252	504,4	5		-1427,60	60,56	-1529,60	76,60	7,14	-1443,40	73,22	-1551,00	77,56	7,45
255	509,0	5		-1044,40	75,38	-1093,60	78,94	4,71	-1018,20	73,76	-1084,20	76,78	6,48
			avg.	**-1147,66**	**65,35**	**-1244,54**	**70,66**	**8,85**	**-1112,21**	**65,86**	**-1247,60**	**70,93**	**12,47**

The constructive heuristic runs typically during a very few milliseconds, and hence most of the computing time is spent in the local search phase of the algorithm. Despite the small value used in our experiments for the maximum computing time, the results show that the variable neighborhood search algorithm can improve significantly the value of the solution. Depending on the quality of the initial solution the percentage of improvement goes up to nearly 43%. This percentage tends to be larger when the level packing procedure is used to build the loading patterns.

Table 2. Computational results with (RBL)

				(FN)					(FR)				
n	M	$inst$	ord	z_{CH}	$\%_{fill}^{CH}$	z_{VNS}	$\%_{fill}^{VNS}$	imp	z_{CH}	$\%_{fill}^{CH}$	z_{VNS}	$\%_{fill}^{VNS}$	imp
15	31,1	10	(OH)	-269,80	48,18	-339,30	61,98	25,76	-278,90	52,02	-339,60	61,92	21,76
20	39,5	10		-385,90	50,06	-460,50	61,63	19,33	-392,00	53,38	-452,70	60,74	15,48
21	39,4	10		-538,00	57,77	-595,70	64,20	10,72	-519,50	53,51	-607,70	67,10	16,98
22	39,4	10		-934,50	57,98	-1049,00	66,27	12,25	-890,50	60,76	-1034,90	63,75	16,22
25	56,0	5		-519,60	56,20	-571,80	67,92	10,05	-386,00	42,70	-552,20	68,46	43,06
29	57,8	10		-880,70	53,33	-1115,20	65,33	26,63	-893,10	56,23	-1087,90	66,37	21,81
30	63,8	5		-472,20	57,54	-552,80	71,52	17,07	-467,80	50,16	-553,00	64,96	18,21
32	62,5	15		-1564,73	54,04	-1828,80	65,40	16,88	-1561,93	55,82	-1881,33	67,51	20,45
35	74,4	5		-570,40	51,24	-648,20	65,74	13,64	-518,20	58,04	-653,40	65,90	26,09
40	79,2	5		-603,20	49,22	-703,00	68,38	16,55	-642,20	53,62	-712,80	67,58	10,99
44	86,2	5		-1243,00	53,34	-1409,40	67,56	13,39	-1119,20	50,90	-1378,20	67,04	23,14
50	105,2	5		-752,80	59,20	-790,80	66,22	5,05	-688,00	56,20	-797,20	67,94	15,87
71	146,0	5		-516,20	49,46	-609,20	68,18	18,02	-545,40	60,78	-622,20	73,04	14,08
75	150,3	20		-1026,05	59,16	-1175,25	67,61	14,54	-986,55	57,35	-1161,60	68,79	17,74
100	204,3	15		-1359,07	56,15	-1553,07	70,36	14,27	-1306,93	57,04	-1540,87	71,47	17,90
120	245,6	5		-2390,20	64,56	-2683,60	71,18	12,28	-2354,00	65,14	-2736,60	69,92	16,25
134	271,4	5		-2449,20	64,56	-2809,20	72,54	14,70	-2310,40	55,76	-2745,00	73,86	18,81
150	294,4	5		-1791,60	60,40	-1965,20	72,54	9,69	-1771,00	57,00	-1955,60	72,36	10,42
199	399,6	15		-2240,87	62,76	-2425,13	72,35	8,22	-2181,33	59,96	-2434,13	73,62	11,59
240	484,8	5		-1124,40	56,52	-1196,40	71,02	6,40	-1135,80	66,26	-1232,00	76,32	8,47
252	504,4	5		-1400,60	56,58	-1497,40	74,96	6,91	-1380,60	64,90	-1507,20	73,92	9,17
255	509,0	5		-1012,20	58,48	-1069,00	76,54	5,61	-974,20	55,86	-1098,40	74,52	12,75
			avg.	**-1092,96**	**56,21**	**-1229,45**	**68,61**	**13,54**	**-1059,25**	**56,52**	**-1231,12**	**68,96**	**17,60**
15	31,1	10	(OA)	-277,20	47,84	-335,20	64,64	20,92	-302,30	56,79	-341,90	64,97	13,10
20	39,5	10		-419,90	57,21	-460,60	66,86	9,69	-405,90	62,27	-452,80	63,39	11,55
21	39,4	10		-530,50	56,60	-601,90	65,96	13,46	-478,90	55,61	-603,90	65,52	26,10
22	39,4	10		-920,00	57,42	-1026,50	64,77	11,58	-805,80	50,90	-1003,20	64,28	24,50
25	56,0	5		-542,40	59,90	-553,20	65,62	1,99	-520,00	60,04	-588,00	67,96	13,08
29	57,8	10		-949,80	61,93	-1136,20	68,34	19,63	-883,30	56,59	-1134,20	66,34	28,40
30	63,8	5		-501,40	61,24	-555,40	68,74	10,77	-428,20	52,26	-546,40	65,58	27,60
32	62,5	15		-1623,93	55,68	-1873,07	67,30	15,34	-1545,73	57,83	-1821,27	67,06	17,83
35	74,4	5		-582,60	55,40	-625,00	69,32	7,28	-518,40	52,38	-646,80	67,54	24,77
40	79,2	5		-576,20	46,54	-712,80	68,70	23,71	-646,40	57,82	-715,20	67,90	10,64
44	86,2	5		-1272,20	58,42	-1452,60	65,58	14,18	-1149,80	51,56	-1451,80	69,58	26,27
50	105,2	5		-727,80	53,86	-806,80	69,18	10,85	-723,80	60,16	-797,80	66,12	10,22
71	146,0	5		-544,80	57,90	-601,40	72,34	10,39	-536,00	60,50	-604,40	70,12	12,76
75	150,3	20		-1038,00	59,69	-1206,15	69,42	16,20	-1042,10	61,34	-1185,75	69,85	13,78
100	204,3	15		-1380,80	60,09	-1561,20	71,57	13,06	-1366,20	61,53	-1470,87	69,83	7,66
120	245,6	5		-2351,40	59,96	-2648,40	72,50	12,63	-2339,00	54,56	-2750,60	72,30	17,60
134	271,4	5		-2582,00	67,64	-2723,40	72,74	5,48	-2280,00	66,50	-2669,20	73,94	17,07
150	294,4	5		-1867,20	64,46	-1990,40	71,82	6,60	-1785,60	68,32	-1951,40	74,38	9,29
199	399,6	15		-2244,80	62,98	-2419,67	74,59	7,79	-2196,80	61,68	-2468,33	74,16	12,36
240	484,8	5		-1161,40	70,04	-1233,80	74,10	6,23	-1102,80	58,10	-1232,40	79,26	11,75
252	504,4	5		-1403,60	58,66	-1500,20	72,10	6,88	-1411,00	62,82	-1531,00	72,40	8,50
255	509,0	5		-1033,20	64,10	-1082,00	76,36	4,72	-994,60	64,56	-1078,20	70,84	8,41
			avg.	**-1115,05**	**58,98**	**-1232,09**	**69,66**	**11,34**	**-1066,48**	**58,82**	**-1229,34**	**69,24**	**16,06**

The best average results are obtained using the strategies (BL), (FN) and (OH). The strategy (FN) that consists in inserting first the client that is nearest to the depot generates usually the best initial solutions when compared to (FR). In some cases, choosing randomly the first client to insert in the route yields better initial solutions, but even in these cases, the local search procedure tends to reach better solutions at the end of the computing time with the strategy (FN) than it does with the strategy (FR). Note that the results remain nearly the same for different runs of the strategy (FR). Ordering the items by height (OH) or by area (OA) has a more significant impact when the level packing procedure is used to place the items in the vehicle. When the bottom-left based strategies (BL) and (RBL) are used, these two orderings yield results that are not significantly different for these instances.

The variants of the algorithm based on the bottom-left placement procedures find solutions with an high percentage of used space in the vehicles. This percentage

Table 3. Computational results with (LP) and $(LP.FF)$

				(FN)					(FR)				
n	M	$inst$	ord	z_{CH}	$\%_{fill}^{CH}$	z_{VNS}	$\%_{fill}^{VNS}$	imp	z_{CH}	$\%_{fill}^{CH}$	z_{VNS}	$\%_{fill}^{VNS}$	imp
15	31,1	10	(OH)	-246,40	39,03	-293,80	51,74	19,24	-256,10	39,35	-293,30	51,00	14,53
20	39,5	10		-332,40	38,31	-394,40	54,39	18,65	-331,50	42,05	-399,70	50,76	20,57
21	39,4	10		-487,50	48,08	-526,70	51,37	8,04	-445,30	42,42	-539,90	54,67	21,24
22	39,4	10		-772,70	42,13	-901,80	50,89	16,71	-773,70	42,51	-897,70	52,67	16,03
25	56,0	5		-437,80	45,26	-475,00	51,22	8,50	-400,60	47,74	-477,20	49,16	19,12
29	57,8	10		-840,70	45,14	-1018,40	52,99	21,14	-816,40	40,43	-973,30	50,08	19,22
30	63,8	5		-431,40	46,90	-501,00	56,42	16,13	-419,00	46,62	-464,40	54,48	10,84
32	62,5	15		-1462,47	46,86	-1554,40	51,45	6,29	-1452,93	46,20	-1631,53	51,47	12,29
35	74,4	5		-532,40	43,68	-587,00	55,40	10,26	-468,20	37,52	-563,60	53,64	20,38
40	79,2	5		-556,60	39,88	-619,40	50,78	11,28	-541,60	46,42	-645,40	54,32	19,17
44	86,2	5		-1113,80	40,56	-1217,60	52,68	9,32	-1018,60	46,16	-1311,20	56,12	28,73
50	105,2	5		-666,60	41,30	-740,40	52,68	11,07	-667,40	41,84	-759,80	54,34	13,84
71	146,0	5		-519,20	49,80	-563,40	58,78	8,51	-477,40	43,30	-555,80	53,96	16,42
75	150,3	20		-984,20	46,16	-1095,85	54,78	11,34	-927,70	43,20	-1087,70	57,22	17,25
100	204,3	15		-1313,40	49,90	-1410,00	58,85	7,35	-1260,87	48,48	-1384,33	55,43	9,79
120	245,6	5		-2302,40	50,54	-2635,80	62,06	14,48	-2223,00	48,96	-2558,60	59,52	15,10
134	271,4	5		-2258,20	55,56	-2510,40	61,70	11,17	-2194,60	52,60	-2392,20	62,02	9,00
150	294,4	5		-1724,00	49,64	-1935,40	62,02	12,26	-1743,60	51,02	-1806,20	64,92	3,59
199	399,6	15		-2180,60	48,18	-2374,33	62,55	8,88	-2179,53	53,97	-2371,60	66,35	8,81
240	484,8	5		-1116,20	55,44	-1192,80	68,30	6,86	-1074,80	48,44	-1188,00	62,98	10,53
252	504,4	5		-1365,80	46,74	-1468,60	70,78	7,53	-1335,60	51,02	-1464,40	63,90	9,64
255	509,0	5		-986,20	50,70	-1039,80	65,70	5,44	-981,40	54,02	-1038,20	63,24	5,79
			avg.	**-1028,68**	**46,35**	**-1138,92**	**57,16**	**11,38**	**-999,54**	**46,10**	**-1127,46**	**56,47**	**14,63**
15	31,1	10	(OA)	-250,22	36,92	-295,78	50,67	18,21	-252,11	36,67	-287,78	46,26	14,15
20	39,5	10		-346,50	37,75	-395,10	50,59	14,03	-339,20	42,76	-412,80	51,44	21,70
21	39,4	10		-486,60	46,81	-538,10	51,53	10,58	-450,80	45,31	-549,70	51,89	21,94
22	39,4	10		-756,20	40,47	-895,30	51,72	18,39	-769,20	43,04	-879,10	48,35	14,29
25	56,0	5		-493,75	41,73	-525,75	51,18	6,48	-423,20	44,20	-476,40	54,86	12,57
29	57,8	10		-965,00	40,43	-1054,75	45,96	9,30	-815,00	45,36	-1026,20	50,45	25,91
30	63,8	5		-430,00	45,14	-490,80	55,38	14,14	-410,40	40,04	-483,20	55,86	17,74
32	62,5	15		-1427,80	42,59	-1581,13	51,43	10,74	-1412,64	43,27	-1683,79	50,77	19,19
35	74,4	5		-532,40	43,68	-591,00	51,36	11,01	-492,20	41,90	-563,00	51,52	14,38
40	79,2	5		-638,75	37,30	-692,25	43,70	8,38	-556,80	42,84	-642,40	55,06	15,37
44	86,2	5		-1121,80	41,66	-1253,60	57,18	11,75	-1067,20	38,30	-1312,00	58,38	22,94
50	105,2	5		-658,80	40,36	-754,20	54,00	14,48	-653,40	47,52	-748,40	55,02	14,54
71	146,0	5		-502,20	46,98	-554,40	54,72	10,39	-497,00	41,76	-555,00	49,10	11,67
75	150,3	20		-1032,58	46,18	-1119,79	53,54	8,45	-924,65	40,84	-1090,60	57,05	17,95
100	204,3	15		-1291,27	48,62	-1463,00	60,69	13,30	-1282,93	47,11	-1422,40	58,85	10,87
120	245,6	5		-2288,00	47,74	-2524,40	61,88	10,33	-2236,20	45,16	-2555,00	61,74	14,26
134	271,4	5		-2309,40	49,20	-2579,20	58,62	11,68	-2555,75	43,00	-2847,50	56,80	11,42
150	294,4	5		-1708,80	49,54	-1871,40	59,36	9,52	-1673,00	44,42	-1898,20	62,82	13,46
199	399,6	15		-2178,00	47,87	-2355,53	61,90	8,15	-2149,33	49,75	-2353,27	62,86	9,49
240	484,8	5		-1378,25	59,38	-1423,75	64,83	3,30	-1086,40	45,42	-1194,40	62,14	9,94
252	504,4	5		-1373,40	48,76	-1442,80	61,60	5,05	-1359,40	51,14	-1477,80	63,08	8,71
255	509,0	5		-974,00	43,44	-1031,00	57,68	5,85	-1194,75	53,03	-1262,75	66,70	5,69
			avg.	**-1051,99**	**44,66**	**-1156,05**	**54,98**	**10,61**	**-1027,34**	**44,22**	**-1169,17**	**55,95**	**14,92**

is typically higher for the largest instances. It goes up to 78.94% when the strategies (BL), (FN) and (OA) are used on the instances with 255 clients and an average of 509 items per instance. The percentage of used space tends to decrease with the level packing procedures. This trend was expectable given that the loading patterns generated through the level packing procedure are more constrained (guillotinable patterns) than those generated with the approaches relying on the bottom-left rules. In general, the results obtained with level packing procedures are outperformed by the bottom-left based approaches.

6 Conclusions

In this paper, we explored the first solution algorithm for the 2L-ESPP. The approach is based on constructive heuristics to generate initial feasible solutions for the problem, and on variable neighborhood search to look for improved incumbents. We described different alternative neighborhood structures based on the

routing and packing characteristics of the solution. We provided also the first results concerning the resolution of this problem for a large set of benchmark instances of the 2L-CVRP. The results illustrate the effectiveness of the variable neighborhood search procedure in improving the solutions of the constructive heuristics. These results allowed the comparison between the different strategies described in this paper. Besides the practical relevance of the problem, these results may contribute for the resolution of the 2L-CVRP through column generation algorithms since the 2L-ESPP is the pricing subproblem that results from the corresponding Dantzig-Wolfe decomposition of this problem.

Acknowledgments. This work was supported by FEDER funding through the Programa Operacional Factores de Competitividade - COMPETE and by national funding through the Portuguese Science and Technology Foundation (FCT) in the scope of the project PTDC/EGE-GES/116676/2010 (Ref. COMPETE: FCOMP-01-0124-FEDER-020430), by FCT within the project scope: UID/CEC/00319/2013, and by FCT through the grant SFRH/BD/73584/ 2010 for Telmo Pinto (funded by QREN - POPH - Typology 4.1 - co-funded by MEC National Funding and the European Social Fund), and by FEDER funds through the Competitiveness Factors Operational Programme - COMPETE.

References

1. Feillet, D., Dejax, P., Gendreau, M., Gueguen, C.: An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. Networks **44**(3), 216–229 (2004)
2. Gendreau, M., Iori, M., Laporte, G., Martello, S.: A Tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints. Networks **51**(1), 4–18 (2008)
3. Hansen, P., Mladenovic, N., Pérez, J.: Variable neighborhood search: methods and applications. Annals of Operations Research **175**, 367–407 (2010)
4. Iori, M., Martello, S.: Routing problems with loading constraints. Top **18**(1), 4–27 (2010)
5. Iori, M., Salazar-Gonzalez, J.J., Vigo, D.: An exact approach for the vehicle routing problem with two-dimensional loading constraints. Transportation Science **41**(2), 253–264 (2007)
6. Mladenovic, N., Hansen, P.: Variable neighborhood search. Computers and Operations Research **24**, 1097–1100 (1997)
7. Pinto, T., Alves, C., de Carvalho, J.V.: Column generation based heuristic for a vehicle routing problem with 2-dimensional loading constraints: a prototype. In: XI Congresso Galego de Estatística e Investigación de Operacións, Spain (2013)
8. Righini, G., Salani, M.: Symmetry helps: bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. Discrete Optimization **3**(3), 255–273 (2006)
9. Righini, G., Salani, M.: New dynamic programming algorithms for the resource constrained elementary shortest path problem. Networks **51**(3), 155–170 (2008)
10. Zachariadis, E.E., Tarantilis, C.D., Kiranoudis, C.T.: A guided tabu search for the vehicle routing problem with two-dimensional loading constraints. European Journal of Operational Research **195**(3), 729–743 (2009)