A Hybrid Heuristic for the 2L-VRP with Clustered Backhauls

Oscar Dominguez[1], Daniel Guimarans[2], and Angel A. Juan[3]

[1] Institute of Intelligent Systems and Numerical Applications in Engineering. University of Las Palmas de Gran Canaria, 35017 Las Palmas de Gran Canaria, Spain
[2] Optimisation Research Group. NICTA. Eveleigh NSW 2015, Australia
[3] Computer Science Dept. - IN3. Open University of Catalonia, 08018 Barcelona, Spain
oscar@opein.com, daniel.guimarans@nicta.com.au, ajuanp@uoc.edu

Abstract. This paper discusses the 2L-VRP with clustered backhauls, a realistic extension of the classical vehicle routing problem where both delivery and pick-up demands are composed of non-stackable items. This problem is frequently found in real-life transportation activities, but it has not been analysed in the literature yet. After describing the problem and reviewing some related work, we propose a hybrid algorithm for solving it. Our approach integrates biased randomization with a metaheuristic framework. Our approach is tested on a set of instances derived from the ones proposed for the 2L-VRP without backhauls.

Keywords: Packing, Routing, Transportation, Vehicle routing problem

1 Introduction

The Vehicle Routing Problem (VRP) is a well-known combinatorial optimisation problem in which a fleet of vehicles has to service a set of customers at the lowest possible cost [14]. The most basic variants of the VRP and richer versions have been extensively studied due to their potential applicability on real transportation activities [2].

In this paper, we consider a realistic variant of the VRP that combines vehicle routing and loading (packing) aspects as well as backhauls. This variant is an extension of the Two-dimensional Capacitated Vehicle Routing Problem (2L-VRP) [8], where customers' demands consist of a set of rectangular items that cannot be stacked due to their weight, dimensions, or fragility. Our work was originally motivated by real-life transportation activities at Opein (www.opein.com), a company which provides industrial equipment to its customers, mostly in the building-construction field. Similar operations appear in other industries where large-sized items pick-up and delivery is also required, e.g. furniture or appliances. These items must be efficiently packed on the truck surface to attain a high vehicle's utilisation. Thus, one needs to consider not only the items weight, but also their length and width. For the purposes of this paper, we consider these items to be of rectangular shape, and we assume they cannot be piled up or overlap.

Several variants of the 2L-VRP have been defined, depending on constraints on the loading configuration: *(i)* two-dimensional sequential oriented loading ($2|SO|L$), where items cannot be rotated nor rearranged en route; *(ii)* two-dimensional sequential non-oriented (rotated) loading ($2|SR|L$), allowing items to be rotated 90° when loaded on the vehicle, but not rearranged; *(iii)* two-dimensional unrestricted oriented loading ($2|UO|L$), where items cannot be rotated but can be rearranged; and *(iv)* two-dimensional unrestricted non-oriented (rotated) loading ($2|UR|L$), allowing items rotation and rearrangement. So far, only Fuellerer et al. [5] have solved all four problem variants. In fact, only Fuellerer et al. [5] and Dominguez et al. [4] have addressed the non-oriented loading configurations. In this work, we consider the oriented case combined with sequential loading ($2|SO|L$). Sequential loading might be a frequent requirement in real-life distribution practices, since unloading and re-loading heavy machinery might represent a significant cost in terms of both time and resources.

As an industrial machinery hire company, Opein also needs to fetch their equipment at the end of the hiring lease. These adds a considerable burden to their logistics and yields separate problems for delivery and pick-ups, dealing with spatial and capacity constraints in both cases. Backhauling has been proven to be an efficient way to achieving significant savings [7]. In the Vehicle Routing Problem with Backhauls (VRPB), the set of customers is divided into delivery locations (*linehaul*) or pick-up points (*backhaul*). The critical assumption is that all deliveries must be made on each route before any pick-ups can be made, as rearrangement of the loads on the trucks at the delivery points is not deemed economical or feasible. This is also coherent with our selection of sequential loading variants of the 2L-VRP. In addition, no route can contain only backhaul customers, although linehaul-only routes are allowed. As customers can be visited only once, all items demanded (linehaul) or supplied (backhaul) by each customer should fit into a single vehicle (item clustering) without surpassing vehicle's capacity and loading surface area.

This article proposes a hybrid algorithm for solving the Two-dimensional Capacitated Vehicle Routing Problem with clustered Backhauls (2L-VRPB), considering sequential loading. Our approach is described in Section 2. To the best of our knowledge, this is one of the first times this problem is tackled in the literature. Only Malapert et al. [11] have previously studied the combination of both problems, but does not provide any computational results. Our proposed method combines a Large Neighbourhood Search (LNS) metaheuristic [12] with biased-randomised versions of classical routing and packing heuristics. Biased randomisation of heuristics refers to the use of skewed probability distributions to induce an oriented (i.e. biased) random behaviour of the heuristic [9]. Our results show that the proposed approach is an efficient way of solving the 2L-VRPB with oriented loading, as presented in Section 3. We finally summarise our main findings and contributions in Section 4.

2 Biased-Randomised Large Neighbourhood Search

Our method combines a Large Neighbourhood Search (LNS) metaheuristic framework [12] with biased-randomised versions of classical routing and packing heuristics. In this framework, an initial solution is gradually improved by alternately destroying and repairing the solution. Typically, LNS requires a reduced number of parameters, thus reducing the need for complex and time-costly tuning processes. Biased randomisation (BR) is included at different stages in our method. Biased randomisation of heuristics refers to the use of skewed probability distributions to transform otherwise deterministic methods into probabilistic algorithms [9]. In our Biased-Randomised Large Neighbourhood Search (BR-LNS) approach, we apply biased-randomised techniques on routing and packing heuristics, as well as in the destruction phase. For the routing component, we used a biased-randomised version of the well-known Clarke and Wright heuristic [3]. As for solving the packing, we use randomised versions of two effective packing heuristics [1, 10]. Our destruction process makes use of splitting techniques similar to those proposed by Juan et al. [9].

We start our algorithm by generating an initial solution by means of the `Pack-and-Route` procedure, explained in detail later. This method provides a complete routing solution with packing plans for both linehaul and backhaul customers. The algorithm then starts an iterative process aiming at improving the best solution by combining partial destruction and reconstruction methods. In our method, a set of adjacent routes is extracted from the current solution using a probabilistic selection process and different geometric criteria [9]. Nodes belonging to these routes are then removed from the solution. This set of customers, together with associated demands, constitutes a sub-instance of the original 2L-VRPB of smaller size, predictably much easier to solve due to the NP-Hard nature of the problem. Sub-problems are also solved by means of the `Pack-and-Route` procedure. The resulting sub-solution is then merged with non-extracted routes to generate a new solution of the original problem. If this new solution improves the incumbent one, we accept it as the new current solution. Finally, the algorithm returns the best found solution when it exhausts its allowed execution time.

The logic for solving the 2L-VRPB is encoded in the `Pack-and-Route` method. As mentioned, we combine biased-randomised versions of well-known heuristics to tackle routing and packing problems in an integrated manner, contrary to most two-stage approaches for the 2L-VRP.

We start our method by generating an initial dummy solution as described in the savings heuristic [3], i.e. we create a return trip from the depot to each customer using as many vehicles as necessary. Next, we compute the savings associated with each edge and sort them in descending order, as specified in the heuristic. In this process, we omit all edges connecting backhaul to linehaul customers, as sequencing restrictions are imposed in the problem. By ruling out backhaul-linehaul edges, we prevent our algorithm to consider forbidden moves, reducing the number of required feasibility checks. Once the savings list is computed, we rearrange its elements applying a biased-randomised process, so edges associated with higher savings are more likely to be ranked at the top of the list. In our case, a geometric probability distribution, driven by a single parameter α ($0 < \alpha < 1$), is used to induce this skewed behaviour. The biased randomisation of the savings list allows edges to be selected in a different way each time the process is called, while maintaining the logic behind the heuristic [9]. At this point, the algorithm starts an iterative route-merging process. At each iteration, the edge at the top of the savings list is selected. If this edge connects a linehaul customer to a backhaul location, and any or both routes containing these customers is already a mixed linehaul-backhaul route, the edge is discarded. Since all backhaul visits should be done after servicing all linehaul customers in a route, only one such edge can be included per route. Otherwise, both routes are merged yielding a new mixed linehaul-backhaul route. In this case, capacity constraints are not considered for the merging, as by definition the vehicle will be empty after visiting all assigned linehaul customers. If the selected edge is a linehaul-to-linehaul or backhaul-to-backhaul link, routes will be merged subject to: *(i)* there is enough weight capacity in the vehicle to carry all items from both routes; and *(ii)* they can be conveniently loaded, i.e. without overlapping and keeping the sequential order defined by the merged route.

Evaluating packing feasibility might become a time-consuming process. In our approach, we use biased-randomised versions of two well-known effective heuristics: Best-Fit [1] and Touching Perimeter [10]. In order to speed up the packing feasibility checking, we use a fast-access memory-based method to determine if we have already computed a packing solution for the same configuration. If the current configuration is already included in the cache memory, routes are merged. Otherwise, we first use the biased-randomised Best-Fit heuristic to compute a packing solution. If the current configuration is yielded as unfeasible by the Best-Fit method, a biased-randomised Touching Perimeter heuristic is called. In both cases, we use a geometric distribution to skew the search, controlled in the two heuristics by a single parameter β ($0 < \beta < 1$). The process is repeated a number of times ($maxPackIter$) before the current configuration is finally disregarded as feasible. Whenever any of the two heuristic methods finds a feasible packing solution, it is stored in the packing cache memory and the process is stopped.

Finally, when a complete solution is computed, we use a fast memory-based technique to possibly improve its associated routes. As in the packing case, we store previously computed routes and packing plans for a given set of customers in a fast-access cache memory. If the obtained solution contains the same set with a higher cost, the route stored in the cache memory is retrieved and the solution is updated. Otherwise, we add (or update) the route to the cache memory for subsequent iterations.

Notice that an important advantage of our approach is its relative simplicity. Our method only uses three parameters, namely α, β, and $maxPackIter$. This significantly reduces the need for fine-tuning processes and its sensitivity to particular problem characteristics, providing a robust method able to perform efficiently across different instances. Moreover, its pseudo-random nature makes it suitable for parallelisation, thus providing high-quality results in low computational times.

3 Computational Experiments

Our algorithm has been implemented in Java. All experiments were run in a standard PC with an Intel Core i3 processor at 3.4 GHz and 8 GB RAM. The algorithm was executed on the Netbeans platform for Java over Windows 7.

As far as we are concerned, we present the first results obtained for the 2L-VRPB. Therefore, we could not find benchmark instances for the problem. Using the method described by Toth and Vigo [13] to generate VRPB instances from classic Euclidean VRP ones, we extended the 2L-VRP benchmark instances [8, 6]. These benchmark sets contain 5 classes with 36 instances each. We have generated two new 2L-VRPB instances for each 2L-VRP one. These new instances contain 50% and 80% linehaul customers, represented by the linehaul/backhaul ratios 1/1 and 4/1, respectively. Tables 1 and 2 present results for the two linehaul/backhaul configurations considering sequential oriented loading ($2|SO|L$). In all cases, we present for each instance and each class the best-found solution obtained with our BR-LNS algorithm and the time required to reach it. As it can be observed, our method is able to solve the proposed benchmark instances in reasonably low computational times. Remarkably, it does so by means of a rather simple approach with very few parameters.

Figure 1 shows a comparison of costs, by class and backhaul configuration level, for the $2|SO|L$. Notice that the solutions obtained in the second configuration scenario are slightly worse than the ones obtained in the first one. In other words: the cost of some instances tends to increase a little bit as configurations with more linehaul customers are considered.

4 Conclusions

This paper presents a hybrid approach for solving the two-dimensional VRP with clustered backhauls (2L-VRPB). This problem can be found in practical transportation applications, such as the one that motivated this work. To the best of our knowledge, this is the first time this realistic extension of the two-dimensional VRP (2L-VRP) is tackled in the literature.

We propose a hybrid algorithm combining Large Neighbourhood Search with biased-randomised versions of well-known routing and packing heuristics. The search is controlled by the LNS framework, which partially destroys a solution by using splitting techniques and rebuilds it with a combination of skewed probability distributions to better guide the search. Packing heuristics are integrated in the routes construction, contrary to most two-stage methods from previous approaches. Our algorithm is also enhanced with memory-based techniques for both routing and packing processes, improving its execution performance without penalising the quality of the generated solutions.

Table 1. Results on the 36 instances of the 2L-VRPB – 1/1 – 2|SO|L

	Class 1		Class 2		Class 3		Class 4		Class 5	
Instance	Sol.	t (s)	Sol.	t (s)	Sol.	t (s)	Sol.	t (s)	Sol.	t (s)
1	301.99	7.34	308.76	0.16	308.76	0.01	312.12	8.53	307.63	0.02
2	308.76	0.01	308.76	0.00	308.76	0.01	308.76	0.01	308.76	0.01
3	335.54	0.00	336.40	0.04	345.66	0.14	335.54	0.01	335.54	0.01
4	375.12	0.00	375.12	0.01	375.12	0.02	375.12	0.00	375.12	0.00
5	372.12	0.01	376.84	0.13	373.71	0.01	372.12	0.02	372.12	0.02
6	432.3	0.01	428.88	0.07	432.30	0.00	432.30	0.02	432.30	0.02
7	689.32	6.30	692.26	0.01	691.85	0.01	699.27	0.00	689.32	3.70
8	689.32	2.52	698.87	0.00	718.89	0.11	692.26	0.24	677.52	3.74
9	494.03	1.50	501.48	0.01	494.03	2.61	500.57	1.39	494.03	9.49
10	502.77	2.42	610.45	0.08	536.29	14.68	589.43	18.85	571.68	23.03
11	502.77	2.18	603.37	0.27	581.42	6.22	644.27	2.46	573.31	26.99
12	471.46	0.00	482.63	0.04	471.46	0.00	475.76	0.06	471.46	0.00
13	2,276.57	4.63	2,399.98	1.28	2,384.40	9.30	2,354.57	10.49	2,326.80	37.73
14	751.69	0.17	870.04	17.33	878.23	15.98	777.60	48.01	771.31	31.59
15	751.69	0.18	850.73	1.00	853.62	20.42	909.02	26.86	907.13	22.48
16	543.09	0.02	549.86	0.69	544.24	17.32	543.39	0.11	542.60	0.17
17	638.14	0.01	635.94	0.11	635.94	0.06	638.14	0.21	635.94	0.15
18	834.86	1.06	937.03	54.06	919.65	67.34	918.57	16.46	845.35	21.74
19	562.83	49.32	655.44	2.79	655.97	35.19	637.33	78.63	617.50	0.16
20	319.72	71.97	419.92	93.00	397.05	2.33	398.26	32.33	375.20	95.47
21	721.78	52.93	876.37	21.22	892.75	85.88	844.89	58.94	783.33	95.90
22	721.68	32.36	872.10	98.03	862.65	44.82	899.02	64.36	805.17	23.92
23	746.9	89.69	880.09	74.67	860.55	30.76	862.22	146.63	802.86	20.87
24	838.96	0.31	920.51	105.86	890.40	36.03	896.58	71.11	844.15	146.21
25	889.59	183.73	1,144.05	20.28	1,102.54	81.73	1,091.96	142.24	984.61	74.50
26	779.21	51.93	1,031.22	36.97	1,039.09	177.26	1,096.63	184.96	903.86	175.61
27	964.88	175.88	1,073.48	157.05	1,089.58	86.97	1,058.67	175.05	1,011.70	142.89
28	1,022.91	215.51	1,780.33	120.89	1,801.48	149.40	1,813.13	233.55	1,616.89	187.81
29	1,217.36	241.36	1,727.00	203.58	1,638.68	245.21	1,667.36	163.74	1,625.58	186.08
30	1,050.11	219.99	1,415.14	101.35	1,396.25	271.84	1,385.71	51.51	1,236.57	229.24
31	1,216.24	233.28	1,686.66	115.50	1,698.68	240.84	1,730.54	265.37	1,545.89	254.49
32	1,202.83	132.09	1,700.82	262.50	1,679.53	270.55	1,687.62	230.04	1,521.70	183.21
33	1,213.71	236.67	1,716.05	272.15	1,715.24	191.94	1,732.86	271.81	1,505.30	270.56
34	702.84	271.66	890.10	236.21	908.90	261.96	877.18	249.25	808.02	247.37
35	747.01	269.40	1,006.72	197.76	1,020.11	245.65	1,027.38	296.90	893.69	264.60
36	488.96	95.28	1,090.58	262.33	1,126.35	251.71	1,052.64	299.11	946.13	265.28

Table 2. Results on the 36 instances of the 2L-VRPB – 4/1 – 2|SO|L

	Class 1		Class 2		Class 3		Class 4		Class 5	
Instance	Sol.	t (s)	Sol.	t (s)	Sol.	t (s)	Sol.	t (s)	Sol.	t (s)
1	259.97	0.00	259.97	0.00	260.22	0.01	275.25	1.50	259.97	0.01
2	299.64	0.00	314.14	0.00	322.42	0.00	299.64	0.00	299.64	0.00
3	349.12	0.00	350.83	0.00	367.86	0.03	356.76	0.04	349.12	0.00
4	415.83	0.00	395.42	0.00	395.42	0.75	410.20	0.01	395.42	0.01
5	376.68	0.00	376.68	0.00	376.68	0.01	385.74	0.01	376.68	0.01
6	432.83	0.00	432.85	0.00	432.83	0.03	432.83	0.06	432.83	0.01
7	598.68	0.00	723.39	1.29	674.70	0.01	674.28	0.73	631.28	3.62
8	598.68	0.00	683.64	0.01	713.49	0.18	660.95	0.01	603.43	0.73
9	571.75	0.18	573.06	0.00	573.06	0.01	571.75	0.41	571.75	0.09
10	512.06	0.02	642.58	4.31	613.95	8.29	663.73	5.52	609.63	1.04
11	512.06	0.13	662.43	2.10	663.37	2.09	737.89	0.34	614.38	4.37
12	523.41	0.01	546.33	0.06	524.53	0.00	534.87	5.43	522.56	0.40
13	1,997.84	0.04	2,489.25	8.45	2,468.80	40.16	2,518.66	1.46	2,286.38	39.29
14	746.28	0.81	1,017.55	3.15	879.84	51.01	900.65	38.14	863.12	30.93
15	746.28	0.80	963.49	4.84	1,024.84	23.10	1,085.14	31.10	1,002.07	8.00
16	613.19	0.01	614.67	0.02	610.99	0.11	622.18	0.70	610.99	0.10
17	725.83	0.01	734.15	0.07	723.17	4.70	724.47	0.43	722.62	0.37
18	791.4	2.34	1,000.84	3.21	971.94	8.75	989.86	86.07	909.63	81.39
19	567.89	0.57	698.50	57.67	742.96	20.53	722.10	68.21	637.06	68.32
20	288.9	94.58	460.16	92.13	466.63	62.67	500.80	97.53	445.67	64.52
21	703.81	111.97	965.26	70.23	1,035.99	51.30	906.63	131.54	848.91	137.90
22	733.42	3.33	990.59	31.10	968.10	134.53	996.26	143.92	888.16	92.15
23	794.85	5.37	958.56	82.44	986.13	10.50	956.42	118.70	873.26	132.09
24	904.53	1.15	1,061.97	60.80	999.72	69.61	1,002.35	122.82	922.58	140.82
25	859.97	4.40	1,312.22	127.47	1,255.11	170.31	1,271.21	168.07	1,088.57	121.26
26	833.59	115.71	1,259.05	54.06	1,229.65	192.96	1,283.55	79.58	1,125.75	186.52
27	1,004.20	130.34	1,245.97	18.04	1,283.22	162.44	1,207.54	176.60	1,149.52	151.64
28	1,059.68	145.08	2,303.88	242.63	2,334.18	231.19	2,186.90	245.29	2,047.11	226.90
29	1,210.77	220.03	2,009.95	165.31	1,986.48	244.40	1,905.98	249.37	1,935.20	248.51
30	1,067.26	52.52	1,687.05	271.49	1,663.83	254.09	1,642.08	260.73	1,445.86	215.40
31	1,260.15	127.00	2,048.22	130.41	2,029.25	257.27	2,141.65	273.80	1,864.63	153.84
32	1,260.15	126.56	2,038.96	273.48	2,026.01	245.78	2,005.18	270.62	1,795.95	250.65
33	1,295.28	118.13	2,081.76	251.38	2,191.51	274.83	2,147.33	273.46	1,880.20	219.55
34	654.71	197.28	1,066.72	185.72	1,087.44	269.66	1,078.34	298.62	946.10	297.71
35	839.02	109.90	1,253.38	266.50	1,308.31	245.42	1,318.93	255.82	1,146.00	187.40
36	584.62	283.14	1,500.73	278.22	1,494.50	298.95	1,448.06	294.29	1,309.89	295.62

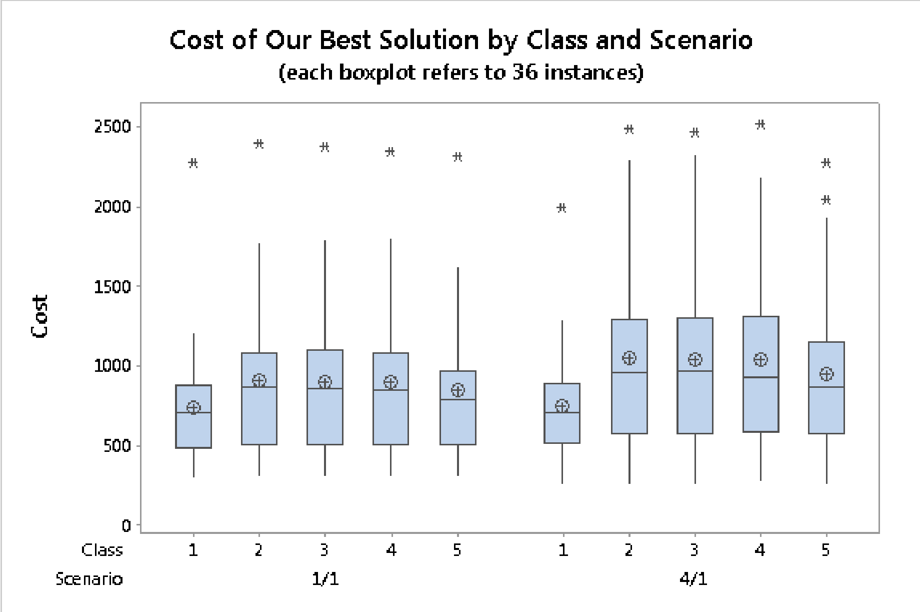

Fig. 1. Comparison of results for the 2L-VRPB with and without items rotation.

We have tested our BR-LNS algorithm in a new set of instances generated for the 2L-VRPB. As our results prove, we are able to solve them in very competitive computational times.

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). NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program.

References

1. Burke, E., Kendall, G., Whitwell, G.: A new placement heuristic for the orthogonal stock-cutting problem. Operations Research 52, 655–671 (2004)
2. Cáceres-Cruz, J., Arias, P., Guimarans, D., Riera, D., Juan, A.: Rich vehicle routing problem: Survey. ACM Computing Surveys 47(2), 1–28 (2014)
3. Clarke, G., Wright, J.: Scheduling of vehicles from a central depot to a number of delivery points. Operations Research 12, 568–581 (1964)
4. Dominguez, O., Juan, A.A., Faulin, J.: A biased-randomized algorithm for the two-dimensional vehicle routing problem with and without item rotations. International Transactions in Operational Research 21, 375–398 (2014)

5. Fuellerer, G., Doerner, K., Hartl, R., Iori, M.: Ant colony optimization for the two-dimensional loading vehicle routing problem. Computers & Operations Research 36, 655–673 (2009)
6. Gendreau, M., Iori, M., Laporte, G., Martello, S.: A tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints. Networks 51, 4–18 (2008)
7. Golden, B., Baker, E., Alfaro, J., Schaffer, J.: The vehicle routing problem with backhauling: two approaches. In: Proceedings of the XXI Annual Meeting of the S.E. TIMS. pp. 90–92. Myrtle Beach, SC (1985)
8. Iori, M., Salazar, J., Vigo, D.: An exact approach for the vehicle routing problem with two-dimensional loading constraints. Transportation Science 41(2), 253–264 (2007)
9. Juan, A., Faulin, J., Jorba, J., Riera, D., Masip, D., Barrios, B.: On the use of Monte Carlo simulation, cache and splitting techniques to improve the Clarke and Wright savings heuristic. Journal of the Operational Research Society 62(6), 1085–1097 (2011)
10. Lodi, A., Martello, S., Vigo, D.: Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems. INFORMS Journal on Computing 11, 345–357 (1999)
11. Malapert, A., Guéret, C., Jussien, N., Langevin, A., Rousseau, L.: Two-dimensional pickup and delivery routing problem with loading constraints. Tech. Rep. CIRRELT-2008-37, CIRRELT, Montréal, Canada (2008)
12. Pisinger, D., Røpke, S.: Large Neighborhood Search. In: Gendreau, M., Potvin, J. (eds.) Handbook of Metaheuristics, pp. 399–419. Springer, New York, NY (2010)
13. Toth, P., Vigo, D.: An exact algorithm for the vehicle routing problem with backhauls. Transportation Science 31, 372–385 (1997)
14. Toth, P., Vigo, D.: Vehicle Routing: Problems, Methods, and Applications. SIAM Publishers, Philadelphia, PA (2014)