Tackling the Loading Aspect of the Vehicle Routing Problem with Compartments

Sandro Pirkwieser[1], Günther R. Raidl[1], Jens Gottlieb[2]

[1] Institute of Computer Graphics and Algorithms
Vienna University of Technology, Vienna, Austria
pirkwieser@ads.tuwien.ac.at, raidl@ads.tuwien.ac.at

[2] SAP AG, Walldorf, Germany
jens.gottlieb@sap.com

Abstract

We propose a new solution approach based on variable neighborhood search for routing problems having vehicles with multiple compartments and where several incompatibilities need to be respected. The introduced additions either directly deal with the packing of the vehicles or try to optimize the routing from a packing viewpoint. Results on a large set of available benchmark instances show the effectiveness of this method.

We investigate the *vehicle routing problem with compartments* (VRPC) which has been tackled in the literature only very recently. We adhere to the definition of a rather general variant given by Derigs et al. [1]; see this original work for more details. In addition to the classical VRP a vehicle has several (at least two) compartments in which the customers' orders have to be placed. As in [1] we will consider the cases of having compartments which are flexible in size/capacity (but bounded by the total vehicle capacity), together with products that are only compatible with specific compartments, as well as fixed compartment capacities and product groups that might not be placed together in the same compartment. The former setting occurs in practice for food retail when delivering frozen and dry goods, whereas the second—and from a computational point of view more challenging and thus interesting—setting occurs when distributing petrol involving different fuel types. In fact, in case of the latter setting, the packing subproblem is NP-hard. In El Fallahi et al. [2] and Muyldermans and Pang [8] a less generic (simpler) scenario was considered, comprising of two compartments with fixed capacities and two product groups, each being compatible with only one compartment. Mendoza et al. [6, 7] tackled the VRPC (which they called the multi-compartment vehicle routing problem (MC-VRP)) with stochastic demands via several construction heuristics and a memetic algorithm. In all variants the customer orders can be split among different vehicles. Contrary to previous work we devoted more effort to the packing aspect in order enhance the overall solution quality and introduce additional suitable neighborhood structures. Besides trying to minimize the total routing costs, which still is the ultimate goal, we also aim at increasing the *density*, i.e. the efficiency, of the packing. This measure is the average squared loading ratio (load divided by capacity) on a per compartment basis for fixed capacities and on a per vehicle basis otherwise. The basic idea is adopted from a concept introduced by Falkenauer and Delchambre for the one-dimensional bin packing and line balancing problem [3]. Opting for a high packing density directly corresponds to maximizing the utilization of the vehicles and hence to enable a more (cost) efficient delivery, e.g. allowing a customer to be visited by only one instead of two vehicles.

Our heuristic solution approach is mainly based on Variable Neighborhood Search (VNS) [4] and includes some of the problem-specific techniques from [1] which were reported to yield good performance. As initial solution for the classical single-trajectory VNS we select the best solution out of several generated with variants of best insertion, the savings algorithm, and the sweep algorithm. In the shaking phase, the core of the VNS, we utilize different move operations. On the one hand, we remove and reinsert orders via choosing them either at random, based on the induced costs (or detour), or on their similarity to a randomly chosen seed order. Whole sets of orders are selected either via considering orders belonging to a certain customer, or being contained in a route which is itself selected at random, having the highest routing costs, or the least density. Similarly, such sets of orders might belong to a randomly selected compartment or the compartment with the least load. On the other hand, we also try to exchange route

n	p	VNS-FF			VNS-BFD		
		%-gap$_{\text{min.cost}}$	%-gap$_{\text{avg.cost}}$	%-gap$_{\text{avg.dens}}$	%-gap$_{\text{min.cost}}$	%-gap$_{\text{avg.cost}}$	%-gap$_{\text{avg.dens}}$
10	2	-0.16	0.35	5.85	-0.16	-0.14	6.86
	3	0.00	0.02	-0.05	0.00	0.00	0.35
25	2	-0.18	0.09	0.57	-0.23	-0.04	1.11
	3	-0.10	0.16	0.18	-0.26	-0.08	0.83
50	2	-0.35	-0.13	0.35	-0.36	-0.12	0.83
	3	-0.12	0.17	0.65	-0.37	-0.11	1.56
100	2	-0.30	-0.04	0.17	-0.24	0.07	0.23
	3	-0.45	-0.23	0.11	-0.75	-0.47	1.27
200	2	-0.23	0.43	-0.86	-0.12	0.28	-0.29
	3	-0.30	-0.04	-0.07	-0.24	-0.02	-0.06
avg.		-0.25	0.03	0.34	-0.34	-0.09	0.93

Table 1: Average results of VNS variants on instances of type food compared to so far best solutions obtained by Derigs et al. [1]

segments of limited size between two different routes. Best-fit and best-fit-decreasing (BFD) strategies are applied when packing a single order and a set of orders, respectively. The insertion in a route's sequence is either done in a purely greedy and thus myopic way or using a regret heuristic [1, 9] which acts more foresighted. To improve upon the actual routing we apply the well-known 3-opt as well as 2-opt* (trying to exchange all routes' end segments) neighborhood structures and the packing (density) is tackled via once reinserting all orders of a route using BFD and iteratively emptying single compartments followed by applying several order exchange moves similarly to the heuristic presented in [5].

The algorithm was implemented in C++, compiled with GCC 4.3 and executed on a single core of a 2.53 GHz Intel Xeon E5540 with 24 GB RAM, 3 GB RAM dedicated per core. For testing we used the instances introduced in [1] and available online at `http://www.ccdss.org/vrp/` together with the best known solutions. Though they did not explicitly state the packing density it can be calculated given the actual assignment of orders to compartments. We consistently set a runtime limit of 10 minutes. The instances differ in type (petrol or food), number of customers (10 to 200, either clustered or not) and products (2 or 3), vehicle capacity (600 to 9000), and maximal order demand. We performed 10 runs per instance and setting and state following results: the minimal and average travel cost as well as the average density as percentage gaps to the so far best known solution; note that in contrast to travel costs a higher density and hence a positive gap is generally better. We considered two variants of the VNS: One which utilizes best-fit, the density measure (including the neighborhoods based on it), and if appropriate the repacking heuristics (VNS-BFD), and another one using first-fit and none of the extensions despite the new neighborhoods not relying on the density or the load (VNS-FF).

The results on the instances of type food are given in Table 1, those on the instances of type petrol are given in Table 2, where we averaged them for instances with the same number of customers n and products p. As expected the VNS benefits more from the extensions for the instances of type petrol. However, also for the food instances where we are faced with a considerably simpler packing subproblem a slight gain can be observed. Altogether, the performance of our algorithmic framework is very encouraging: For 145 out of 200 instances a new best solution could be obtained by VNS-BFD, the same objective value was reached for 32 instances, and only for 23 instances the solution quality is inferior. Remarkably, VNS-FF performs very similar with respect to the new best solutions, but a Wilcoxon rank sum test with an error level of 5% revealed that VNS-BFD is in total 43 times significantly better (on 13 food and 30 petrol instances) and only 15 times worse (on 7 food and 8 petrol instances).

Since for the instances of type petrol the NP-hard packing subproblem was so far only solved heuristically, we checked the feasibility when inserting several orders at once also in an exact way. However, for none of these instances an additional packing could be found. Also because of this we decided to modify the available instances to exhibit a more challenging packing aspect, i.e. having less well-formed

n	p	VNS-FF			VNS-BFD		
		%-gap$_{\text{min.cost}}$	%-gap$_{\text{avg.cost}}$	%-gap$_{\text{avg.dens}}$	%-gap$_{\text{min.cost}}$	%-gap$_{\text{avg.cost}}$	%-gap$_{\text{avg.dens}}$
10	2	0.00	0.08	-0.42	0.00	0.02	1.98
	3	0.00	0.37	0.04	0.00	0.39	0.79
25	2	-0.14	0.11	-0.18	-0.15	-0.01	0.90
	3	-0.37	0.57	-1.68	-0.45	0.43	-0.30
50	2	-0.56	-0.01	0.46	-0.54	-0.18	1.24
	3	-0.70	0.16	0.22	-0.77	0.01	0.70
100	2	-0.56	0.10	0.74	-0.87	-0.26	1.87
	3	-0.90	-0.06	1.98	-1.22	-0.27	2.29
200	2	-0.74	-0.31	-0.62	-1.47	-0.97	1.77
	3	-3.34	-1.63	2.50	-3.67	-2.32	4.79
avg.		-0.62	0.06	0.26	-0.77	-0.16	1.29

Table 2: Average results of VNS variants on instances of type petrol compared to so far best solutions obtained by Derigs et al. [1]

order demands. First results indeed show more clearly a gain due to the extensions, and also when exactly solving some packings a small but sometimes significant gain is observable this time.

References

[1] Ulrich Derigs, Jens Gottlieb, Jochen Kalkoff, Michael Piesche, Franz Rothlauf, and Ulrich Vogel. Vehicle routing with compartments: applications, modelling and heuristics. *OR Spectrum*, 2010. Available online, DOI: 10.1007/s00291-010-0194-3.

[2] Abdellah El Fallahi, Christian Prins, and Roberto Wolfler Calvo. A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. *Computers & Operations Research*, 35:1725–1741, 2008.

[3] Emanuel Falkenauer and Alain Delchambre. A genetic algorithm for bin packing and line balancing. In *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, volume 2, pages 1186–1192, May 1992.

[4] Pierre Hansen, Nenad Mladenović, Jack Brimberg, and José A. Moreno Pérez. Variable neighborhood search. In Michel Gendreau and Jean-Yves Potvin, editors, *Handbook of Metaheuristics, 2nd edition*, chapter 3, pages 61–86. Springer, 2010.

[5] John Levine and Frederick Ducatelle. Ant colony optimization and local search for bin packing and cutting stock problems. *Journal of the Operational Research Society*, 55:705–716, 2004.

[6] Jorge E. Mendoza, Bruno Castanier, Christelle Guéret, Andrés L. Medaglia, and Nubia Velasco. Constructive heuristics for the multi-compartment vehicle routing problem with stochastic demands. *Transportation Science*, 2010. (Forthcoming).

[7] Jorge E. Mendoza, Bruno Castanier, Christelle Guéret, Andrés L. Medaglia, and Nubia Velasco. A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands. *Computers & Operations Research*, 37(11):1886–1898, 2010.

[8] Luc Muyldermans and Gu Pang. On the benefits of co-collection: Experiments with a multi-compartment vehicle routing algorithm. *European Journal of Operational Research*, 206:93–103, 2010.

[9] David Pisinger and Stefan Ropke. A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34(8):2403–2435, 2007.