Vehicle Routeing with Multiple Use of Vehicles — Source link

Éric D. Taillard, Gilbert Laporte, Michel Gendreau

Institutions: Université de Montréal

Published on: 01 Aug 1996 - Journal of the Operational Research Society (Palgrave Macmillan UK)

Topics: Vehicle routing problem and Tabu search

Published in *The Journal of the Operational Research Society*, vol. 47, no. 8, pp. 1065-1070, which should be cited to refer to this work.
DOI: 10.2307/3010414

CRT-95-19

VEHICLE ROUTING WITH MULTIPLE USE OF VEHICLES

by

Éric D. Taillard[1]
Gilbert Laporte[1]
and
Michel Gendreau[1,2]

[1] Centre de recherche sur les transports, Université de Montréal, Case postale 6128, succursale Centre-ville, Montréal, CANADA H3C 3J7
[2] Département d'informatique et de recherche opérationnelle, Case postale 6128, succursale Centre-ville, Montréal, CANADA H3C 3J7

Centre de recherche sur les transports — Publication CRT-95-19
Département d'informatique et de recherche opérationnelle — Publication #964

March 1995

ABSTRACT

The Vehicle routing problem with multiple use of vehicles is a variant of the standard vehicle routing problem in which the same vehicle may be assigned to several routes during a given planning period. A tabu search heuristic is developed for this problem. It is shown to produce high quality solutions on a series of test problems.

Key words : Vehicle routing problem, multiple use of vehicles, heuristic, tabu search.

RÉSUMÉ

Le problème de tournées de véhicules avec utilisations multiples des véhicules est une variante du problème de tournées de véhicules standard dans lequel le même véhicule peut être réutilisé pour plusieurs tournées au cours d'un horizon de planification donné. On développe un algorithme de recherche avec tabous pour ce problème. L'algorithme permet d'obtenir des solutions de très bonne qualité sur un ensemble de problèmes tests.

Mots-clefs : Problème de tournées de véhicules, utilisation multiple de véhicules, heuristique, recherche avec tabous.

Introduction

The *Vehicle Routing Problem* (VRP) is a central problem in distribution management. Its most standard version can be formally defined as follows. Let $G = (V, E)$ be an undirected graph where $V = \{v_0, v_1, ..., v_n\}$ is a set of vertices representing *cities* or *customers*, and $E = \{(v_i, v_j) \mid v_i, v_j \in V, i < j\}$ is the edge set. Vertex v_0 denotes a depot at which are based m identical vehicles of capacity Q, where m is a decision variable or a constant. Each city of $V \setminus \{v_0\}$ has a non-negative demand q_i and a non-negative service time s_i. A distance matrix (c_{ij}) is defined on E. Here, we use the terms distance and travel time interchangeably. The VRP consits of designing a set of m vehicles routes having a minimum total length and such that 1) each route starts and ends at the depot, 2) each remaining city is visited exactly once by one vehicle, 3) the total demand of a route does not exceed Q, 4) the total duration (including service and travel times) of a route does not exceed a preset limit L.

The VRP is a hard combinatorial optimization problem for which several exact and approximate algorithms have been designed (see, e.g., Laporte[1]). A number of meaningful variants and extensions have also been analyzed (see, e.g., Assad[2] as well as Laporte and Osman[3]).

One drawback of the standard VRP definition is that it implicitly assumes each vehicle is used only once over a planning period of duration M. For example, M could correspond to an eight hour working day. In several contexts, once the vehicle routes have been designed, it may be possible to assign several of them to the same vehicle and thus use fewer vehicles. When m is given *a priori* and Q is relatively small, this will often be the only practical option. However, this possibility is not directly accounted for in the problem statement and more often than not, an efficient “packing” of the routes into working days will be hard to achieve. Designing routes with multiple uses of the vehicles is rather important in practice, but this problem (denoted by the abbreviation VRPM) has received very little attention in the Operational Research literature. To our knowledge, only Fleischmann[4] has explicity addressed this problem. In his working paper, this author proposes a savings based heuristic for the VRPM and illustrates it on examples involving between 68 and 361 customers.

We believe the VRPM deserves more attention. In this article, we propose a new heuristic for this problem. The algorithm and computational results are presented in the following two sections, and the conclusion follows.

Algorithm

In recent years, several powerful tabu search algorithms have been proposed for the VRP (see, e.g., Taillard[5], Gendreau, Hertz and Laporte[6], Rochat and Taillard[7]). As a rule, these algorithms produce very good and sometimes optimal solutions. There are, however, cases where the search becomes trapped into a local optimum and standard diversification techniques are not powerful enough to counter this situation. The major interest of the Rochat and Taillard[7] algorithm is that it allows diversification of the search process to take place by generating and combining promising solutions, not unlike what is done in genetic algorithms[8,9]. More precisely, the route generation procedure first produces several good VRP solutions using tabu search. It then extracts single vehicle routes from this population of solutions, and combines some of these routes to define a partial starting solution for another application of tabu search. This process is repeated a number of times and some of the vehicle routes generated are selected as candidates for the final VRP solution. Note that each application of tabu search has the effect of producing a full VRP solution starting from a limited set of routes and it may also modify these seed routes through the local search process. We now provide a description of the algorithm we have designed for the VRPM, based on the Rochat-Taillard principle.

The proposed heuristic is made up of three parts. It first generates a large set of good vehicle routes satisfying the VRP constraints. It then makes a selection of a subset of these routes using an enumerative algorithm. Finally, it assembles the selected routes into feasible working days using several applications of a bin packing heuristic. The idea of first generating individual routes and combining them into a global solution has already been implemented by a number of authors (see, e.g., Foster and Ryan[10], Ryan, Hjorring and Glover[11], Renaud, Boctor and Laporte[12], Rochat and Taillard[7]). Compared with previous work, the parallel algorithm of Rochat and Taillard produces a much broader set of routes and the average quality of these routes is also higher. Using bin packing in the final part of the algorithm is a natural choice and has already been suggested by Fleischmann[4]. Before providing a step by step description of the algorithm, we should mention that it can also easily handle a variant of the VRPM where penalties are incurred for overtime, as is often the case in practice.

PART 1. Route generation procedure

Step 1. (First set of VRP solutions) Generate 20 VRP solutions with an unspecified number of vehicles, using the tabu search algorithm of Taillard[5]. Insert the individual vehicle routes in a *list*, and label each of them by the value of the VRP solution.

Step 2. (Generation of new VRP solutions). Apply the following operations p times, where p is an input parameter:

i) Randomly select a route from the list according to a criterion that gives a larger weight to routes that are often generated or that belong to better VRP solutions.

ii) Disregard all routes having vertices in common with the already selected routes; if some routes remain, go to i).

iii) Using the routes selected in i) as a starting point, apply tabu search to generate a new VRP solution, append the individual routes to the list, and label them as in Step 1. Dominated routes are eliminated. If a route is duplicated, only one copy is kept, but the frequency of that route is recorded as this affects its probability of being selected.

PART 2. Generation of VRP solutions

At most q routes are selected for the second part of the algorithm, where q is an input parameter and $q >> m$. Typically, the number of selected routes is q, but there can be fewer if the size of the list is less than q. Routes are selected in non-decreasing order of their labels and inserted in a set J. This selection rule is such that a feasible VRP solution can always be obtained. Then, within a search tree, all feasible VRP solutions that can possibly be constructed by combining routes of J are generated. In order to control the growth of the search tree, branching priority is always given to routes containing the largest number of customers. This process ends with a set K of feasible VRP solutions.

PART 3. Generation of solutions to the VRPM

In the last part of the algorithm, an attempt is made to obtain a feasible solution to the VRPM by solving a packing problem for each VRP solution of K, and selecting the best overall solution. For each VRP solution k of K, let $f_{k,\ell}$ be the duration of the ℓ^{th} route, where $\ell = 1, ..., m_k$, and m_k is the number of routes in solution k. Then, for each k, a VRPM solution is identified whenever there exists a feasible bin packing[13] solution with m identical bins of size M, and m_k items of weights $f_{k,1}, f_{k,2}, ..., f_{k,m_k}$. To identify such bin packing solutions, all items are first sorted in non-increasing order of weights and gradually assigned to the bin of least accumulated weight. If none of the m bins has a weight exceeding M, this procedure terminates. Otherwise, an attempt is made to obtain a feasible solution by repeatedly swapping items belonging to different bins. Feasibility at this stage is not guaranteed. When overtime is permitted, any hour worked beyond time M is penalized by a factor θ (e.g., when $\theta = 0.5$, overtime is paid at 50% over the standard rate). In our implementation, we used $\theta = 2$. In such a case, the procedure just described can be applied, with the exception that the swapping step attempts to minimize total overtime. Here, a feasible VRPM solution always exists.

Computational results

The algorithm just described was tested on a number of VRPM instances generated as follows. We first used the same graphs, demands and vehicle capacities as in problems 1–5 and 11–12 of Christofides, Mingozzi and Toth[14], and problems 11–12 of Fisher[15]. Starting from these nine base problems, several instances were generated by using different values of m and M. For each value of m, two values of M were used: $M_1 = [1.05\ z^*/m]$ and $M_2 = [1.1\ z^*/m]$, where $[x]$ is the value of x rounded to the nearest integer and z^* is the value of a VRP solution obtained as in Rochat and Taillard[7] with an unspecified number of vehicles. Larger values of m were not considered as these produced infeasible instances (with $2c_{oi} > M$, for some i). The main characteristics of the base problems are summarized in Table 1.

Table 1 Characteristics of the base problems

Problem number	Source	n	m	z^*
1	CMT-1[1]	50	1,...,4	524.61
2	CMT-2[1]	75	1,...,7	835.26
3	CMT-3[1]	100	1,...,6	826.14
4	CMT-4[1]	150	1,...,8	1028.42
5	CMT-5[1]	199	1,...,10	1291.44
6	CMT-11[1]	120	1,...,5	1042.11
7	CMT-12[1]	100	1,...,6	819.56
8	F-11[2]	71	1,...,3	241.97
9	F-12[2]	134	1,...,3	1162.96

[1] CMT: Christofides, Mingozzi and Toth[14]; [2] Fisher[15].

Table 2 Summary of computational results

Problem number	n	m	$\leq M_1$	$\leq M_2$	$\lvert J\rvert$	$\lvert J'\rvert$	$\lvert K\rvert$	$\lvert K'\rvert$	Time (minutes)
1	50	1 2 3 4	** ** — —	** ** — *	62.5	171	25.8	136	5
2	75	1 2 3 4 5 6 7	** ** ** ** ** — —	** ** ** ** ** * —	219.8	500	838.0	8 615	7
3	100	1 2 3 4 5 6	** ** ** ** — —	** ** ** ** * *	393.4	500	453.0	3 053	24
4	150	1 2 3 4 5 6 7 8	** ** ** ** ** * — —	** ** ** ** ** ** * —	400	500	777.8	33 551	51
5	199	1 2 3 4 5 6 7 8 9 10	** ** ** ** ** ** ** ** * —	** ** ** ** ** ** * ** ** *	400	500	10000.6	45 138	66
6	120	1 2 3 4 5	** ** * — *	** ** ** * **	365.0	500	112.8	400	45
7	100	1 2 3 4 5 6	** ** ** ** — —	** ** ** ** ** *	149.2	500	233.2	24 360	23
8	71	1 2 3	** — —	** ** —	81	255	36.8	194	26
9	134	1 2 3	** ** **	** ** **	308.0	500	176.8	2 191	75

The algorithm was coded in Pascal and run on a 100 Mhz Silicon Graphics Indigo machine. Each instance was run five times, using different random seeds in the local search algorithm for the VRP solutions. The main computational results are summarized in Table 2. The column headings are as follows:

- Problem number (as in Table 1).
- n: number of customers.
- m: number of vehicles allowed in the final solution.
- $\leq M_1, \leq M_2$: In these two columns, ** means that each of the five runs produced a feasible solution where the workload of each vehicle did not exceed M_1 or M_2 (here we used $q = 400$ in Part 1 of the algorithm). Sometimes, at least one run did not produce such a feasible solution; then the VRP solutions obtained after Part 1 of the five runs were pooled together in the hope of identifying a feasible solution and the algorithm was rerun with $q = 500$ in Part 1; * means that this additional run produced a feasible solution; – means that it failed.
- $|J|$: average number of vehicle routes retained in Part 2 of the algorithm; this average is computed over the first five runs (with $q = 400$).
- $|J'|$: number of vehicle routes retained at in Part 2 when a sixth run was made (with $q = 500$).
- $|K|$: average number of VRP solutions produced with J.
- $|K'|$: number of VRP solutions produced with J'.
- Time: average CPU time in minutes over all runs.

Results presented in Table 2 indicate that the algorithm successfully produced feasible solutions within reasonable computing times for most instances. For M_1, the success rate was 34/52 (for **) and 38/52 (for * or **); for M_2, the corresponding ratios are 40/52 and 44/52. Because of the way the instances were generated, these results indicate that the feasible solutions are within

5% or 10% of the best known VRP solution. When executing our tests, we have observed that similar results are obtained if about half the current computation time is used. Another observation stemming from Table 2 is that the number of single vehicle routes of full VRP solutions tends to grow with n. This implies the algorithm has a higher likelihood of finding a feasible VRPM solution when n is large.

Table 3 Computational results for instances infeasible with respect to M_1

				M_1			M_2		
Problem number	n	m	z^*	Length	Cost	Longest route ratio	Length	Cost	Longest route ratio
1	50	3	524.61	533.00	579.48	1.115	529.17	556.51	1.050
		4		546.29	565.27	1.027	546.29	546.29	0.985
2	75	6	835.26	841.60	857.19	1.032	839.22	839.22	0.996
		7		843.60	878.29	1.073	843.60	849.74	1.023
3	100	5	826.14	829.50	853.23	1.062	829.50	833.04	1.010
		6		842.85	861.18	1.032	832.00	838.07	1.012
4	150	7	1028.42	1042.39	1074.16	1.033	1031.43	1035.03	1.010
		8		1049.02	1088.03	1.075	1044.98	1059.99	1.029
5	199	10	1291.44	1316.00	1331.09	1.024	1316.00	1316.00	0.981
6	120	4	1042.11	1042.11	1055.07	1.020	1042.11	1092.11	0.973
7	100	5	819.56	819.56	836.80	1.050	819.56	820.77	1.003
		6		819.56	845.48	1.064	819.56	823.79	1.014
8	71	2	241.97	241.97	249.97	1.031	241.97	241.97	0.985
		3		244.60	257.31	1.075	244.60	299.31	1.027

We then present in Table 3 results concerning the instances that were infeasible for M_1. The first two column headings are identical to those of Table 2. We then report for each of M_1 and M_2 the following statistics:

– z^*: value of the VRP solution obtained using the Rochat and Taillard algorithm[7].

– Length: total length of the best solution identified for the VRPM.

– Cost: total cost of that solution, i.e. total length, plus penalty $(\theta = 2)$ associated with routes whose length exceeds M.

– Longests route ratio: length of the longest route divided by M.

Table 3 sheds more light on the quality of the results. Comparing the z^* and length columns shows that our solution costs are on the average within 1.2% of z^*. As the z^* values are believed to be quasi-optimal for the VRP, this means our VRPM solution should also be close to optimality. The "largest route ratio" columns indicate that the longest route is almost always within 10% of the allowed limit, and this value is below 5% in half of the cases with M_1, and below 3% in all cases with M_2. These values seem to make sense from a practical point of view and confirm the quality of the proposed approach.

Conclusion

We have considered a practical and difficult variant of the VRP in which vehicles can be used several times during a given planning period. This problem had previously received very little attention in the Operational Research literature. We have proposed an efficient and robust heuristic that produces high quality solutions on all instances that were attempted. We believe our approach could easily be applied to solve real-life problems.

Acknowledgements

This work was partially supported by the Canadian Natural Sciences and Engineering Research Council under grants OGP0038816 and OGP0039682. Éric Taillard also benefitted from an NSERC International postdoctoral fellowship and was partically funded by Strategic grant STR0149269. This support is gratefully acknowledged.

References

1. G. Laporte (1992). The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research* 59, 345–358.
2. A.A. Assad (1988). Modeling and implementation issues in vehicle routing. In *Vehicle Routing: Methods and Studies*. (B.L. Golden and A.A. Assad, Eds) pp. 7–45. North-Holland, Amsterdam.
3. G. Laporte and I.H. Osman (1995). Routing problems: A bibliography. *Annals of Operations Research*. Forthcoming.
4. B. Fleischmann (1990). The vehicle routing problem with multiple use of vehicles. Working paper, Fachbereich Wirtschaftswissenschaften, Universität Hamburg.
5. É. Taillard (1993). Parallel iterative search methods for vehicle routing problems. *Networks* 23, 661–676.
6. M. Gendreau, A. Hertz and G. Laporte (1994). A tabu search heuristic for the vehicle routing problem. *Management Science* 40, 1276–1290.
7. Y. Rochat and É. Taillard (1995). Probabilistic Diversification and Intensification in Local Search for Vehicle Routing. Publication CRT-95–13, Centre de recherche sur les transports, Montreal.
8. J.H. Holland (1976). *Adaptations in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, Michigan.
9. J.-Y. Potvin and S. Bengio (1994). A genetic approach to the vehicle routing problem with time windows. Publication CRT-953, Centre de recherche sur les transports, Montreal.
10. B.A. Foster and D.M. Ryan (1976). An integer programming approach to the vehicle scheduling problem. *Operational Research Quarterly* 27, 367–384.

11. D.M. Ryan, C. Hjorring and F. Glover (1993). Extension of the petal method for vehicle routing. *Journal of the Operational Research Society* 44, 289–296.

12. J. Renaud, F.F. Boctor and G. Laporte (1994). An improved sweep heuristic for the vehicle routing problem. Publication CRT-983, Centre de recherche sur les transports, Montreal.

13. S. Martello and P. Toth (1990). *Knapsack Problems*. Wiley, Chichester.

14. N. Christofides, A. Mingozzi and P. Toth (1979). The vehicle routing problem. In *Combinatorial Optimization*. (N. Christofides, A. Mingozzi, P. Toth and C. Sandi, Eds) pp. 313–338. Wiley, Chichester.

15. M.L. Fisher (1994). Optimal solution of vehicle routing problems using minimum K-trees. *Operations Research* 42, 626–642.