Improved Tabu Search Algorithm for the Handling of Route Duration Constraints in Vehicle Routing Problems with Time Windows

Author(s): J. -F. Cordeau, G. Laporte and A. Mercier
Source: *The Journal of the Operational Research Society*, Vol. 55, No. 5 (May, 2004), pp. 542-546
Published by: Palgrave Macmillan Journals on behalf of the Operational Research Society
Stable URL: http://www.jstor.org/stable/4101915
Accessed: 26-05-2016 13:41 UTC

Journal of the Operational Research Society (2004) 55, 542–546 0160-5682/04 $25.00

www.palgrave-journals.com/jors

Improved tabu search algorithm for the handling of route duration constraints in vehicle routing problems with time windows

J-F Cordeau*[1], G Laporte[1] and A Mercier[2]

[1]Canada Research Chair in Distribution Management, HEC Montréal, Canada; and [2]GERAD, École Polytechnique de Montréal, Canada

This note introduces a refinement to a previously proposed tabu search algorithm for vehicle routing problems with time windows. This refinement yields new best known solutions on a set of benchmark instances of the multi-depot, the periodic and the site-dependent vehicle routing problems with time windows.

Journal of the Operational Research Society (2004) **55,** 542–546. doi:10.1057/palgrave.jors.2601707

Keywords: tabu search; multi-depot; periodic; site-dependent vehicle routing; time windows

Introduction

In a previous article,[1] the authors have introduced a unified tabu search heuristic for vehicle routing problems with time windows. This heuristic was applied successfully to the vehicle routing problem with time windows (VRPTW) and two of its generalizations: the periodic VRPTW (PVRPTW) and the multi-depot VRPTW (MDVRPTW). It was also later applied to the site-dependent VRPTW (SDVRPTW) by Cordeau and Laporte.[2] These four problems are natural extensions of the classical vehicle routing problem.[3] In the VRPTW,[4] each delivery must take place within a prespecified time window. The PVRPTW operates on a horizon of several days. Customers specify acceptable combinations of delivery days. The problem is to simultaneously select such a combination for each customer and plan delivery routes. In the MDVRPTW, vehicles are based at one of several depots as opposed to only one in the VRPTW. Finally, in the SDVRPTW several vehicle types are considered and each customer can only be visited by a subset of vehicle types.

The search mechanism used by this algorithm consistently produces high-quality feasible solutions. In particular, each customer visit takes place within its time window, but otherwise no particular care is paid to the scheduling aspect of the problem. The purpose of this note is to show that in the presence of route duration constraints, important gains can be achieved by better incorporating scheduling decisions within the search process. The paper is organized as follows. In the next section we recall the definitions of the problems considered in this study. We then describe the enhancement brought to our algorithm. This is followed by computational results and by the conclusion.

*Correspondence: J-F Cordeau, Canada Research Chair in Distribution Management, HEC Montréal, 3000, Chemin de la Cote-Sainte-Catherine, Montréal, QC, Canada H3T 2A7.
E-mail: jean-francois.cordeau@hec.ca

Vehicle routing problem with time windows

The VRPTW is defined on a complete graph $G=(V, A)$, where $V=\{v_0, v_1,\dots,v_n\}$ is the vertex set and $A=\{(v_i, v_j): v_i, v_j \in V, i \neq j\}$ is the arc set. Vertex v_0 represents a depot at which is based a fleet of m vehicles, and the remaining vertices of V represent customers to be serviced. With each vertex $v_i \in V$ are associated a non-negative load q_i (with $q_0=0$), a non-negative service duration d_i (with $d_0=0$) and a time window $[e_i, l_i]$, where e_i and l_i are non-negative integers. Each arc (v_i, v_j) has non-negative cost c_{ij} and travel time t_{ij}. The VRPTW consists of designing m vehicle routes on G such that (i) every route starts and ends at the depot; (ii) every customer belongs to exactly one route; (iii) the total load and duration of route k do not exceed Q_k and D_k, respectively; (iv) the service at customer i begins in the interval $[e_i, l_i]$, and every vehicle leaves the depot and returns to the depot in the interval $[e_0, l_0]$; and (v) the total travel time of all vehicles is minimized.

The PVRPTW generalizes the VRPTW by considering a planning horizon of several days during which each customer must receive a certain number of visits. The MDVRPTW also generalizes the VRPTW by considering multiple depots at which the vehicles are based. Finally, the SDVRPTW introduces vehicles of different types and compatibility constraints between the vehicles and the customers. As explained by Cordeau *et al*[1] and Cordeau and Laporte,[2] the MDVRPTW and the SDVRPTW can be seen as special cases of the PVRPTW. The same solution approach can thus be used to address all three variants.

An improved tabu search algorithm

The tabu search heuristic that we have developed addresses all problems with a unified methodology. This heuristic moves, at each iteration κ, from the current solution s to the best non-tabu solution s' in the neighbourhood $N(s)$ of s. Let S denote the search space, $c(s)$ denote the cost of a solution $s \in S$ and s^* denote the best feasible solution found during the course of the search. The heuristic can then be summarized as follows:

(1) Construct a random initial solution $s \in S$. If s is feasible, set $s^* := s$ and $c(s^*) := c(s)$; otherwise, set $c(s^*) := \infty$.
(2) For $\kappa = 1, \ldots, \eta$, do
   a. Choose a solution $s' \in N(s)$ that minimizes $c(s')$ and is not tabu.
   b. If s' is feasible and $c(s') < c(s^*)$, set $s^* := s'$ and $c(s^*) := c(s')$.
   c. Set $s := s'$.
(3) Return s^*.

An important feature of the heuristic is that it allows infeasible solutions during the course of the search. The evaluation of candidate solutions in $N(s)$ involves intensive computations since the schedule of the vehicle routes involved in the exchange must be updated to evaluate the impact on violations of time window and route duration constraints in Step 2a of the algorithm. To simplify this evaluation, we have made the assumption that the service at the first customer visited by a vehicle starts as early as possible, that is, at the lower bound of the customer's time window or as soon as the vehicle arrives from the depot. This rule may, however, lead to a violation of the route duration constraint whereas a different scheduling would yield a feasible route.

Consider a particular ordered route $k = (v_0, \ldots, v_i, \ldots, v_q)$ where v_0 and v_q both represent the depot. Let A_i, W_i and B_i denote the arrival time, the waiting time and the beginning of service at vertex v_i, respectively. It is clear that sequentially setting $B_0 = e_0$ and $B_i = \max\{e_i, A_i\}$ for $i = 1, \ldots, q$ is optimal in terms of minimizing time window violations because the vehicle leaves the depot as early as possible and the service at each vertex also begins as early as possible. However, because of route duration constraints, a solution that is infeasible if $B_0 = e_0$ and $B_i = \max\{A_i, e_i\}$ for every vertex v_i can in fact be feasible (or less infeasible) if the departure from the depot is voluntarily delayed. Of course, a simple adjustment that will reduce route duration is obtained by setting $B_0 = \max\{e_0, e_j - t_{0j}\}$, where v_j denotes the first vertex visited after leaving the depot. However, it may sometimes be possible to further delay the departure from the depot, especially when the time window associated with vertex v_j is wide.

Savelsbergh[5] has introduced the concept of forward time slack that may be used to postpone the beginning of service at a given node without causing any time window violation. Assuming $d_i = 0$ for every node i, the author defines the *forward time slack* F_i of vertex v_i as

$$F_i = \min_{i \leqslant j \leqslant q} \left\{ l_j - \left(B_i + \sum_{i \leqslant p < j} t_{p,p+1} \right) \right\} \tag{1}$$

Using the fact that

$$B_j = B_i + \sum_{i \leqslant p < j} t_{p,p+1} + \sum_{i < p \leqslant j} W_p \tag{2}$$

one can rewrite (1) as

$$F_i = \min_{i \leqslant j \leqslant q} \left\{ l_j - \left(B_j - \sum_{i < p \leqslant j} W_p \right) \right\} \tag{3}$$

$$= \min_{i \leqslant j \leqslant q} \left\{ \sum_{i < p \leqslant j} W_p + (l_j - B_j) \right\} \tag{4}$$

The latter form emphasizes the fact that the slack at vertex v_j is the cumulative waiting time up to vertex v_j, plus the difference between the end of the time window and the beginning of service at vertex v_j. It also generalizes directly to the case of non-zero service times.

When feasibility must be maintained through exchanges, the forward time slack is the largest increase in the beginning of service at vertex v_i that will not cause any time window violation. In our case, since infeasible solutions are allowed during the search, the notion of forward time slack must be slightly modified to represent the largest increase in the beginning of service at vertex v_i that will not cause any increase in time window violations. Hence, the term $(l_j - B_j)$ should be replaced with $(l_j - B_j)^+$ in (4) because even if the time window for vertex v_j cannot be satisfied in the current route, one can nevertheless increase the beginning of service at vertex v_i by as much as $\sum_{i<p\leqslant j} W_p$ without increasing the violation of the time window constraint at vertex v_j.

As a result, Equation (4) becomes

$$F_i = \min_{i \leqslant j \leqslant q} \left\{ \sum_{i < p \leqslant j} W_p + (l_j - B_j)^+ \right\} \tag{5}$$

Setting $B_0 = e_0 + F_0$ instead of $B_0 = e_0$ will thus yield a modified route of minimal total duration with equal violations of time window constraints. Observe that delaying the departure time from the depot by $\sum_{0<p<q} W_p$ does not affect the arrival time A_q at the end of the route, whereas delaying the departure by more would simply increase A_q by as much. As a result, the minimal route duration that does not increase constraint violations is given by

$$A_q - \left(e_0 + \min\left\{ F_0, \sum_{0<p<q} W_p \right\} \right)$$

We have thus modified our algorithm so as to compute the forward time slack at the depot whenever the cost of an exchange is evaluated in Step 2a of the algorithm. This obviously leads to an increase in computation time, but it also yields much improved solutions.

Computational results

In this section, we report results on three sets of instances that incorporate route duration constraints. These sets contain MDVRPTW, PVRPTW and SDVRPTW instances that were introduced in two previous papers.[1,2] To our knowledge, no other results have been reported on these instances, except for a recent article on the MDVRPTW by Polacek *et al.*[6] These authors have obtained solutions that improve upon those published by Cordeau *et al*,[1] but are dominated by those of the present study.

Tables 1–3 provide a summary of the computational experiments performed with our new algorithm. In these tables, the value of t denotes the number of depots in the case of the MDVRPTW, the number of days in the case of the PVRPTW and the number of vehicle types in the case of the SDVRPTW. Depending on the problem type, the value of m denotes the number of vehicles available at each depot, on each day or of each type. For each instance, we report the best solution identified after 10^4, 10^5 and 10^6 iterations of a single run of the algorithm. We also indicate the new best solution found during 10 runs with 10^6 iterations and the old best known solution that was reported by Cordeau *et al*[1] or Cordeau and Laporte.[2] Finally, we indicate the CPU time in minutes needed to perform 10^4 iterations on a 2 GHz Pentium 4 computer. Since the time needed to compute an initial solution is negligible, the total computing time is directly proportional to the number of iterations. The CPU time for performing 10^5 or 10^6 iterations is thus obtained by multiplying the reported time by 10 or 100.

A comparison with the CPU times needed by our previous implementation shows that the extra time spent for the computation of the forward time slack yields approximately a two-fold increase in computational effort. However, solution quality improves dramatically. Using the improved algorithm, the average cost of the best known solutions for the MDVRPTW, PVRPTW and SDVRPTW decreases by 2.52, 3.16 and 2.05%, respectively. For each instance of each problem, we have obtained a solution that improves or matches the previous best known solution. Furthermore, solutions found after only 10^5 iterations are very often better than the previous best known solution. Of course, the impact of computing the forward time slack varies from one instance to another. In some cases, this impact is negligible because several customers have time windows whose lower bound is close to the start of the planning horizon. On other instances, however, the impact is significant. This is the case, for example, for instance 6a of the MDVRPTW where the gain exceeds 7%.

Conclusion

We have introduced a simple refinement to a previously proposed tabu search heuristic for several classes of vehicle routing problems with time windows. This refinement is an adaptation of the forward time slack concept previously

Table 1 Computational results for the MDVRPTW

	n	t	m	*Time (10^4)*	*10^4 iter.*	*10^5 iter.*	*10^6 iter.*	*New best*	*Old best*
1a	48	4	2	0.28	1074.12	1074.12	1074.12	1074.12	1083.98
2a	96	4	3	0.79	1776.07	1766.94	1762.48	1762.21	1763.07
3a	144	4	4	1.15	2423.13	2420.89	2397.06	2373.65	2408.42
4a	192	4	5	1.44	2924.15	2868.64	2865.71	2852.29	2958.23
5a	240	4	6	1.81	3092.21	3059.40	3050.80	3029.65	3134.04
6a	288	4	7	2.21	3888.13	3701.08	3670.13	3627.18	3904.07
7a	72	6	2	0.53	1433.35	1425.87	1418.22	1418.22	1423.35
8a	144	6	3	1.02	2167.48	2118.50	2118.50	2102.61	2150.22
9a	216	6	4	1.60	2788.76	2777.91	2760.46	2737.82	2833.80
10a	288	6	5	2.27	3585.85	3546.24	3507.26	3505.27	3717.22
1b	48	4	1	0.32	1037.33	1025.14	1016.59	1005.73	1031.49
2b	96	4	2	0.81	1503.64	1486.26	1486.26	1478.51	1500.48
3b	144	4	3	1.43	2121.87	2033.75	2028.85	2011.24	2020.58
4b	192	4	4	1.88	2270.96	2228.64	2228.64	2202.08	2247.72
5b	240	4	5	2.27	2565.78	2555.95	2527.60	2494.57	2509.75
6b	288	4	6	2.61	3048.46	2978.60	2960.93	2901.02	2943.90
7b	72	6	1	0.61	1364.16	1250.18	1241.25	1236.24	1250.09
8b	144	6	2	1.46	1911.77	1870.34	1823.24	1792.61	1809.35
9b	216	6	3	2.62	2386.30	2338.74	2288.38	2285.10	2310.92
10b	288	6	4	2.63	3283.23	3147.79	3120.32	3079.16	3131.90
Avg.				1.49	2332.34	2283.75	2267.34	2248.46	2306.63

Table 2 Computational results for the PVRPTW

	n	*t*	*m*	*Time (10^4)*	*10^4 iter*	*10^5 iter.*	*10^6 Iter.*	*New best*	*Old best*
1a	48	4	3	0.30	2934.70	2915.68	2915.58	2911.03	3007.84
2a	96	4	6	0.70	5096.18	5094.39	5071.84	5055.05	5328.33
3a	144	4	9	1.09	7378.56	7284.32	7243.79	7229.73	7397.10
4a	192	4	12	1.55	8196.88	8087.06	8087.06	7953.08	8376.95
5a	240	4	15	1.89	8856.69	8752.72	8595.53	8593.00	8967.90
6a	288	4	18	2.45	11 307.65	10 961.78	10 961.78	10 927.45	11 686.91
7a	72	6	5	0.53	6927.63	6891.76	6833.21	6825.07	6991.54
8a	144	6	10	1.14	10 137.66	9990.46	9944.27	9748.36	10 045.05
9a	216	6	15	2.06	13 973.60	13 796.75	13 635.64	13 614.47	14 294.97
10a	288	6	20	2.90	18 235.55	18 135.60	17 769.04	17 735.59	18 609.72
1b	48	4	3	0.37	2373.64	2297.21	2294.03	2294.03	2318.37
2b	96	4	6	0.78	4378.29	4335.11	4335.11	4257.40	4276.13
3b	144	4	9	1.20	5909.71	5699.78	5648.76	5648.76	5702.07
4b	192	4	12	1.90	6838.41	6619.56	6594.54	6594.54	6789.73
5b	240	4	15	2.22	7426.21	7138.28	7138.28	7054.95	7102.36
6b	288	4	18	2.93	9155.26	9039.29	8966.80	8928.37	9180.15
7b	72	6	4	0.73	5696.81	5580.22	5505.23	5505.23	5606.08
8b	144	6	8	1.48	8129.83	7914.39	7875.43	7875.31	7987.64
9b	216	6	12	2.53	11 620.24	11 269.13	11 067.64	10 889.77	11 089.91
10b	288	6	16	3.18	14 339.35	14 145.37	14 031.94	13 980.55	14 207.64
Avg.				1.60	8445.64	8297.44	8225.78	8181.09	8448.32

Table 3 Computational results for the SDVRPTW

	n	*t*	*m*	*Time (10^4)*	*10^4 iter.*	*10^5 iter.*	*10^6 iter.*	*New best*	*Old best*
1a	48	4	2	0.27	1673.34	1666.47	1656.22	1655.42	1655.42
2a	96	4	3	0.49	2925.36	2915.55	2906.13	2904.13	2975.18
3a	144	4	4	0.95	3406.21	3406.21	3324.79	3321.44	3373.52
4a	192	4	5	1.28	4586.40	4532.58	4511.45	4509.36	4682.84
5a	240	4	6	1.73	6195.05	5859.03	5810.85	5777.56	6278.89
6a	288	4	7	2.15	6230.15	5773.24	5769.87	5769.87	6014.95
7a	72	6	2	0.28	2264.87	2181.77	2179.39	2167.46	2179.06
8a	144	6	3	0.83	4029.05	3983.08	3959.06	3904.84	3988.18
9a	216	6	4	1.39	5102.62	5008.54	4922.22	4875.62	4982.19
10a	288	6	5	2.10	6212.78	6171.16	6038.46	5969.50	6050.41
1b	48	4	2	0.31	1438.39	1433.24	1429.35	1429.35	1429.35
2b	96	4	3	0.69	2550.87	2516.83	2502.25	2494.53	2494.54
3b	144	4	4	1.23	2856.24	2814.61	2798.46	2798.46	2807.69
4b	192	4	5	1.57	3896.00	3762.38	3716.41	3711.96	3775.86
5b	240	4	6	2.21	4955.04	4955.04	4672.41	4672.41	4721.91
6b	288	4	7	2.79	5191.36	5008.27	4993.57	4873.08	4934.40
7b	72	6	2	0.41	1872.98	1864.11	1852.70	1837.94	1850.44
8b	144	6	3	1.07	3273.36	3215.06	3186.49	3170.46	3175.25
9b	216	6	4	1.90	4173.86	4033.63	4016.34	3985.52	4017.27
10b	288	6	5	2.90	5316.80	5158.89	5158.89	5147.68	5159.84
Avg.				1.33	3907.54	3812.99	3770.27	3748.83	3827.36

introduced by Savelsbergh. Computational experiments show that in the presence of route duration constraints, the extra computation time required is largely justified by very significant improvements in solution quality.

References

1 Cordeau JF, Laporte G and Mercier A (2001). A unified tabu search heuristic for vehicle routing problems with time windows. *J Opl Res Soc* **52**: 928–936.

2 Cordeau JF and Laporte G (2001). A tabu search algorithm for the site dependent vehicle routing problem with time windows. *INFOR* **39**: 292–298.

3 Toth P and Vigo D (2002). *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications, SIAM Publishing: Philadelphia, PA.

4 Cordeau JF *et al* (2002). VRP with time windows. In: Toth P and Vigo D (eds) *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications. SIAM Publishing: Philadelphia, PA, pp 157–193.

5 Savelsbergh M (1992). The vehicle routing problem with time windows: minimizing route duration. *ORSA J Comput* **4**: 146–154.
6 Polacek M, Hartl R, Doerner K and Reimann M (2003). *A variable neighborhood search for the multi depot vehicle routing problem with time windows*. Technical report, Department of Production and Operations Management, Institute of Management Science, University of Vienna.

Received September 2003; accepted November 2003 after one revision