Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computers & Operations Research 33 (2006) 1129–1137

computers & operations research

www.elsevier.com/locate/cor

Vehicle routing and scheduling with dynamic travel times

Jean-Yves Potvin[a, b, *], Ying Xu[a, b], Ilham Benyahia[c]

[a]*Centre de recherche sur les transports, Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Qué., Canada H3C 3J7*

[b]*Département d'informatique et de recherche opérationnelle, Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Qué., Canada H3C 3J7*

[c]*Département d'informatique et d'ingénierie, Université du Québec en Outaouais, C.P. 1250, succursale B, Hull, Qué., Canada J8X 3X7*

Available online 25 November 2004

Abstract

The field of dynamic vehicle routing and scheduling is growing at a fast pace nowadays, due to many potential applications in courier services, emergency services, truckload and less-than-truckload trucking, and many others. In this paper, a dynamic vehicle routing and scheduling problem with time windows is described where both real-time customer requests and dynamic travel times are considered. Different reactive dispatching strategies are defined and compared through the setting of a single "tolerance" parameter. The results show that some tolerance to deviations with the current planned solution usually leads to better solutions.

Keywords: Transportation; Routing; Dynamic; Heuristics; Tolerance

1. Introduction

Real-time vehicle routing and scheduling has been the subject of many studies during the last few years, as reviewed in [1–3]. Different questions and issues arising in a dynamic context are also discussed in [4,5]. With regard to the work done in this field, our main contribution is the introduction of a "tolerance" concept. In particular, it is empirically demonstrated that reactive dispatching strategies based on some tolerance to deviations to the current planned routes (due to dynamic events) leads to better overall results

* Corresponding author. Centre de recherche sur les transports, Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Qué., Canada H3C 3J7.

0305-0548/$ - see front matter
doi:10.1016/j.cor.2004.09.015

when compared with simpler strategies, such as immediate reaction to any deviation or no reaction at all. In this paper, dynamic events that relate both to the occurrence of new customer requests and dynamic travel times are considered.

The remainder of the paper is the following. In Section 2, a static version of the vehicle routing and scheduling problem is presented. This is followed in Section 3 by a description of a dynamic environment where new customer requests occur in real-time and where travel times are subjected to stochastic variations. Then, a dispatching algorithm is developed in Section 4 to handle both types of events. In the case of dynamic travel times, a tolerance to deviations is proposed to guide the course of actions when a vehicle does not show according to the planned schedule. Computational results in Section 5 provide an indication of the benefits associated with the latter strategy. Concluding remarks follow in Section 6.

2. The static problem

The problem is motivated from the management of local routes in international courier services. This is a many-to-one type of problems where express mail is collected at different customer locations and brought back to a central depot, for further shipping. The static version of the problem can be characterized as an uncapacitated vehicle routing problem with time windows (VRPTW), which is formally stated in the following.

Graph: We have a complete graph $G=(V, E)$ where $V=\{0, 1, 2, \ldots, n\}$ is the vertex set and E is the edge set. Vertices $i=1, \ldots, n$ correspond to customers, whereas vertex 0 is the depot. A non negative travel time t_{ij} is associated with each edge $(i, j) \in E$.

Customers: Each customer i is characterized by a pick-up location, a service time s_i, a time window $[e_i, l_i]$ and a vehicle planned arrival time t_i. If $t_i < e_i$, the vehicle has to wait up to e_i before servicing the customer. If $t_i > l_i$, a penalty is incurred in the objective.

Depot: The depot is characterized by a location, a time window $[e_0, l_0]$ for vehicle arrivals and departures, as well as a vehicle return time t_0^k for each vehicle $k \in K$, where K is the set of vehicles. The service time at the depot is assumed to be $s_0=0$.

Vehicles: We have a fixed number of identical uncapacitated vehicles, where each vehicle, if used, travels along a single route that starts and ends at the depot.

Objective: The objective is to minimize, over all vehicles, a weighted summation of (1) travel time, (2) sum of lateness at customer locations and (3) lateness at the depot. Given a solution $S=\bigcup_{k\in K} S^k$, where $S^k=\{i_0^k, i_1^k, \ldots, i_{m_k}^k\}$ is the sequence of customer locations visited by vehicle k, with $i_0^k=i_{m_k}^k=0$, the objective function can be expressed as follows:

$$\begin{aligned} f(S) &= \sum_{k\in K} f(S^k) \\ &= \sum_{k\in K}\left(\alpha_1 \sum_{p=1}^{m_k} t_{i_{p-1}^k, i_p^k} + \alpha_2 \sum_{p=1}^{m_{k-1}} (t_{i_p^k} - l_{i_p^k})^+ + \alpha_3 (t_0^k - l_0)^+\right), \end{aligned}$$

where α_1, α_2 and α_3 are weighting parameters and $(x-y)^+=\max\{0, x-y\}$.

3. The dynamic problem

A dynamic version of the problem presented in Section 2 is now considered. The dynamic elements include both new request arrivals and dynamic travel times. These will be described below, followed by a description of the system's operating mode.

3.1. Dynamic elements

Customer requests: New requests continuously occur during the day and must be inserted at least cost into the current planned routes.

Travel times: Three different elements are considered to set the travel time between two customer locations:

- *Long-term forecasts*: These travel time forecasts represent well documented long-term trends. Although not dynamic, as they are known well in advance and are not subject to stochastic variations, these forecasts are time-dependent and vary depending on the time period (e.g., morning, lunch time, afternoon).
- *Short-term forecasts*: When the vehicle is ready to depart from its current customer location, the travel time to its next destination is modified by a random uniform amount (positive or negative). This modification, which implies a rescheduling of the planned route, represents short-term forecasts or nowcasts based on any new information available at this time.
- *Dynamic perturbation*: The travel time to the next destination is finally perturbed by adding a value generated with a normal probability law of mean 0 (and different standard deviations, see Section 5). This perturbation represents any unforeseen events that may occur along the current travel leg and represents the truly dynamic component of the travel time. This perturbation is known to the dispatching system only when the vehicle arrives at its planned destination.

The time-dependent travel times associated with long-term forecasts are addressed in [6]. Basically, the scheduling horizon is partitioned into a number of time periods and the travel times change from one period to the next (see [7,8] for alternative models). When the vehicle travels along a link, the travel time calculation is adjusted each time a boundary between two consecutive time periods is crossed. In this way, a vehicle that leaves earlier will arrive earlier at destination, which is known as the "first-in-first-out" (FIFO) property. Details about these time-dependent calculations and the FIFO property can be found in [6]. The short-term forecast and dynamic perturbation are then added to obtain the "true" arrival time at destination. However, only the planned arrival time at destination, based on the long-term and short-term forecasts, is assumed to be known when the vehicle departs.

3.2. Operating mode

A number of assumptions about the system's operating mode, which impact the algorithmic developments presented in the next section, are introduced here.

- A number of requests that have been received at the end of the day (but too late to be serviced the same day) are scheduled during the night for the next day. These requests are said to be "static" as they are used to construct in advance a set of initial planned routes.

- A central dispatching office receives the customer calls and manages the planned routes.
- Communication between the dispatch office, drivers and customers takes place at customer locations.
- Drivers are informed of their next destination when they depart from a customer location (i.e., they have no knowledge of their current planned route).

This simplified model was used here to study the impact of different reactive strategies on solution quality. Eventually, some of these assumptions could be relaxed or modified to obtain a more realistic model. For example, one could assume that communication can take place at any time between the dispatch office and the drivers. But this, in turn, would lead to new issues that are difficult to address, with or without dynamic travel times (like opportunities to divert vehicles away from their current destination, see [9]).

4. Dispatching algorithm

An insertion heuristic is used to dispatch requests to vehicle routes, followed by a local improvement procedure. In the next subsections, we first describe how the initial routes are generated using the set of static requests. Then, the discrete-event simulation scheme used to handle dynamic events is described.

4.1. Initial solution using static requests

The algorithm for generating initial routes considers the static customer requests one by one, in random order, and finds the insertion place with minimum additional cost.

Let S be the current solution and V_s be the set of static customer requests. Using $\leftarrow$ for the assignment operator and Ø for the empty set, this insertion algorithm can be written as follows:

1. $S \leftarrow \emptyset$;
2. while $V_s \neq \emptyset$ do:

   2.1 $s \leftarrow$ random selection in V_s;
   2.2 for each insertion place in S, find the one that minimizes $\Delta f = f(S+s) - f(S)$ and insert s at this place;
   2.3 $V_s \leftarrow V_s - \{s\}$.

As the number of vehicles is fixed, there may be one or more vehicles with empty routes. Accordingly, the insertion of customer s in some empty route is also considered in step 2.2. The notation $f(S+s)$ simply represents the solution cost after the insertion of customer s.

Once the insertion phase is completed, a reoptimization phase takes place. First, a local descent based on CROSS exchanges is applied [10]. Here, two segments of routes are exchanged between two different routes (by removing two arcs in each route and by appropriately reconnecting the two segments). At each iteration, the CROSS exchange that leads to the largest improvement to the current solution is selected. This is repeated until no more improvement can be obtained. At the end, each modified route is further improved with a local descent that considers the relocation of each customer at another place in the same route.

4.2. Solution evolution using dynamic requests

Once the initial static routes are generated, the simulation of the operations day can start. During the simulation, different types of events trigger different types of responses. These events are described below, using the following notation for each vehicle $k \in K$.

- i^k the current destination;
- j^k the next customer location to be serviced (which may be different from i^k, see below);
- $(j+1)^k$ the successor of j^k;
- TTL^k the tolerance time limit. If TTL^k is exceeded, j^k is removed from the route of vehicle k and reassigned to some other route (see below).

Occurrence of a new request: The new customer request s is inserted at minimum additional cost into one of the planned routes (including empty routes), using the insertion procedure described in Section 4.1. If we assume that the chosen insertion place is in the route of vehicle k, the following steps are performed:

1. $S^k \leftarrow S^k + s$;
2. update the planned arrival time of s and all following locations in S^k.

Then, the reoptimization procedure of Section 4.1 is applied, by considering moves that involve route k (or any other route modified during the local search).

TTL^k is reached for some $k \in K$: The tolerance time limit on some vehicle route has been reached. Consequently, customer j^k is removed from the route of vehicle k and inserted at minimum additional cost into the route of another vehicle k'. Note that vehicle k must still reach its current destination i^k, without servicing it, before going to the next location $(j+1)^k$. More precisely, the following steps are performed:

1. Update route of vehicle k.

   1.1. $S^k \leftarrow S^k - \{j^k\}$;
   1.2. $j^k \leftarrow (j+1)^k$; (j^k is now different from i^k);
   1.3. $TTL^k \leftarrow (t_{j^k} + t_f) - t_{i^k,j^k}$.

2. Update route of vehicle k'.

   2.1 $S^{k'} \leftarrow S^{k'} + j^k$;
   2.2 update the planned arrival time of j^k and all following locations in $S^{k'}$.

Then, the reoptimization procedure of Section 4.1 is applied, by considering moves that involve route k' (or any other route modified during the local search process).

In this algorithmic description, parameter t_f corresponds to the maximum delay, with regard to the vehicle planned arrival, which can be tolerated before taking a reassignment action. Fig. 1 provides an illustrative example. Each customer location in this figure is labeled with the corresponding planned arrival time. We assume here that every customer is currently serviced within its time window (so, there

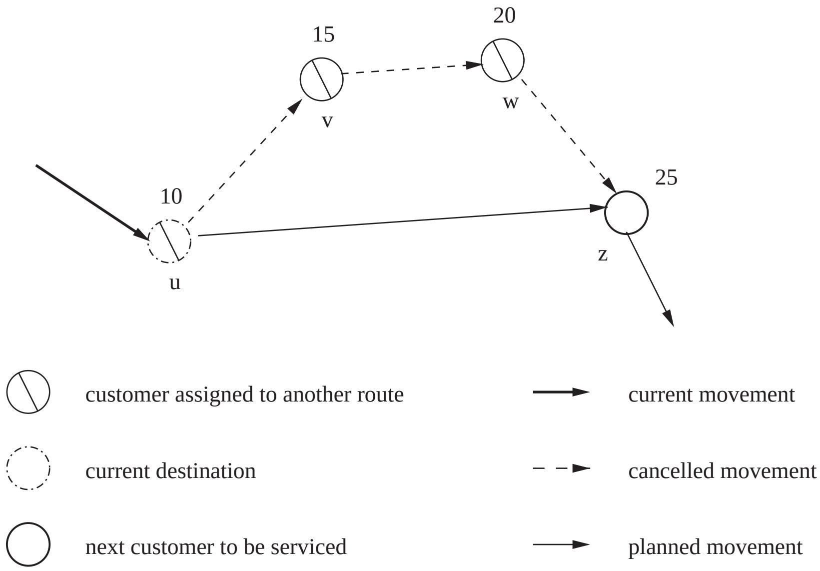

Fig. 1. Reassignment of customers.

is no waiting time) and that the travel time between any two customer locations is 5 time units. The vehicle is currently moving towards customer u and is expected to reach it at time 10. The following customers v, w, z will be visited at time 15, 20 and 25, respectively. Assuming that t_f is set to 1, customer u will be moved to another route if the vehicle does not arrive by the time $10+1=11$. Customer v will also be moved to another route at the same time. If the vehicle has still not reached location u at time $(20+1)-5=16$, then customer w will also be moved, as it is not possible for the vehicle to reach w before time $20+1=21$. The situation illustrated in the figure would thus occur at any time t, $16<t\leqslant 21$ (with $u=i^k$ and $z=j^k$). After time 21, customer z would also be moved to another route. Parameter t_f is important as it leads to different ways of handling dynamic travel times. When $t_f=0$, the reaction is immediate: an action is taken as soon as the vehicle does not arrive at the planned time. At the other end of the spectrum, $t_f=\infty$ means that no action is taken (i.e., we simply wait for the vehicle; no customers are reassigned to other routes).

Note that the reassignment (from vehicle k to vehicle k') of the customer associated with location i^k can be canceled, if vehicle k actually arrives at i^k before vehicle k'. If, at this time, vehicle k' has not yet reached the predecessor of i^k in its current planned route, i^k is simply removed from it. Otherwise, vehicle k' reaches i^k without servicing it. In Fig. 1, customer $i^k=u$ may thus still be serviced by the same vehicle, if the latter arrives at u before the vehicle to which u has been reassigned.

Vehicle arrival at current destination: When the vehicle actual arrival time is known, the route schedule is updated. The response to this event is described below in the “standard situation” with no cancelation of a previous customer reassignment.

1. if $(i^k=j^k)$ then

   1.1 $S^k \leftarrow S^k-\{j^k\}$;
   1.2 $j^k \leftarrow (j+1)^k$;

2. if the actual arrival time at i^k is different from t_{i^k} then update the planned arrival times in S^k;
3. $TTL^k \leftarrow \infty$.

Vehicle departure: Just before departure from the current location, the short-term forecast is introduced to calculate the travel time to the next customer location and the route schedule is updated accordingly. The TTL of the vehicle route is also calculated. More precisely:

1. $i^k \leftarrow j^k$;
2. introduce the short term forecast into the travel time and update the planned arrival times in S^k;
3. $TTL^k \leftarrow t_{j^k} + t_f$.

In the following section, we analyze the impact of various tolerance levels t_f on solution quality.

5. Computational results

For testing purposes, Solomon's VRPTW benchmark problems were used. These 100-customer Euclidean problems are divided into six different classes, depending on the spatial distribution of customers and length of the scheduling horizon. In the problems of type C (clustered), the customers are grouped into well-defined geographic clusters; in the problems of type R (Random), they are randomly distributed; and in the problems of type RC, both clustered and randomly generated customer locations are found. Here, the focus is on the problem classes C2, R2 and RC2, which are particularly interesting for this study because the time horizon is larger and more customers are scheduled on each route. In these problems, the travel times correspond to Euclidean distances and are equal to 40 on average, with a standard deviation of 12.

Half of the customers were used to generate the initial set of routes. The occurrence of each remaining (dynamic) customer i was generated at time $e_i * r$, where r is a random number uniformly distributed between 0 and 1. Three different time periods of equal length were defined (morning, lunch time, afternoon). Time-dependency was obtained by multiplying the travel times by coefficients 1.25, 0.5 and 1.25, respectively. Thus, the vehicles travel faster during lunch time. The short-term bias was obtained by perturbing the coefficients with a random value uniformly chosen in the interval $[-0.1, +0.1]$. The dynamic perturbations were generated according to a normal law of mean 0, with small $(\sigma = 1, 4)$ or large $(\sigma = 16, 32)$ standard deviations. In this study, only delays to the current schedule were considered. Thus, a negative value was simply reset to 0 (no perturbation).

The number of vehicles, for each problem instance, was set to the number of routes in the best solution reported in the literature (which is between 2 and 4, see http://w.cba.neu.edu/∼msolomon/heuristi.htm for the exact numbers). Different settings for the tolerance level t_f were considered, ranging from $t_f = 0$ to $t_f = \infty$. In the first case, the reassignment of the next customer to be serviced is performed as soon as the vehicle does not show at the planned arrival time. In the second case, nothing is done, as we always wait for the vehicle. Intermediate values correspond to different levels of tolerance (i.e., we are willing to wait for the vehicle, but only to a certain extent).

In Tables 1–3, the solution values correspond to $f(S)$ with $\alpha_1 = \alpha_2 = \alpha_3 = 1$ (see Section 2). Each entry in these tables is an average taken over 11, 8 and 8 problem instances, respectively. Each entry in a given table is also normalized using the best average obtained over all σ and t_f values. The numbers in bold correspond to the best results obtained with a given value of σ over all t_f values.

Table 1
Problem class R2

t_f	$\sigma = 1$	$\sigma = 4$	$\sigma = 16$	$\sigma = 32$
0	2.018	2.016	3.882	4.683
0.25σ	1.609	1.589	2.876	3.864
0.5σ	1.445	1.555	**2.750**	**3.800**
σ	1.266	1.400	3.555	5.524
2σ	1.018	1.254	4.241	6.766
3σ	1.010	**1.238**	4.632	6.826
4σ	**1.000**	1.239	4.600	6.808
∞	1.006	1.247	4.649	6.962

Table 2
Problem class C2

t_f	$\sigma = 1$	$\sigma = 4$	$\sigma = 16$	$\sigma = 32$
0	3.182	3.036	4.292	5.210
0.25σ	2.350	2.754	3.630	4.333
0.5σ	1.865	2.402	2.409	**2.220**
σ	1.498	1.829	**1.805**	2.405
2σ	1.086	1.276	1.906	2.585
3σ	1.022	1.182	1.992	2.698
4σ	**1.000**	**1.171**	2.214	2.750
∞	1.012	1.184	2.200	2.725

Table 3
Problem class RC2

t_f	$\sigma = 1$	$\sigma = 4$	$\sigma = 16$	$\sigma = 32$
0	1.723	1.791	2.884	3.339
0.25σ	1.237	1.682	2.595	2.951
0.5σ	1.025	1.627	2.500	**2.854**
σ	1.233	1.265	**2.156**	3.014
2σ	1.084	1.233	2.832	3.876
3σ	1.010	**1.208**	3.213	4.231
4σ	**1.000**	1.254	3.344	4.200
∞	1.005	1.246	3.445	4.115

These results first show that problems associated with larger σ values are more difficult to solve (as the magnitude of the dynamic perturbations increases). For problems with $\sigma = 1, 4$, the best results are obtained with $t_f = 3\sigma, 4\sigma$. Thus, it is generally better to wait for the vehicle's arrival, given that events with a magnitude larger than three or four standard deviations are unlikely to occur (similarly, note that the small variations observed between $t_f = 4\sigma$ and ∞ are basically due to randomness). The picture is quite different with $\sigma = 16, 32$. Although there is some variation with regard to the best t_f value, the latter is always less than or equal to σ. In this case, the reassignment strategy is applied in a non-negligible number of cases and with substantial benefits, if one considers the degradation observed when no such reassignment takes place (c.f., $t_f = \infty$). It is thus a good strategy to wait a little to "catch" events of small magnitude, and to react only to events of larger magnitude.

6. Conclusion

This paper empirically demonstrates that some tolerance to unforeseen delays is indicated for better solutions to emerge. Clearly, more sophisticated optimization techniques, like metaheuristics, could be considered to generate even better solutions. In our case, the computation times of our local search procedures were negligible (when a new dynamic request is inserted, the current solution does not change much and the search converges within a fraction of a second to a local minimum). This could be quite different with a metaheuristic, leading to considerations about CPU time/solution quality trade-offs. Also, more sophisticated ways of setting the tolerance factor t_f could be devised. For example, different classes of customers might lead to different tolerances. Similarly, the tolerance on routes with more slack time should probably be larger than on "tight" routes.

References

[1] Dror M, Powell WB, editors. Special issue on stochastic and dynamic models in transportation. Operations Research 1993;41:1–235.
[2] Gendreau M, Potvin J-Y. Dynamic vehicle routing and dispatching. In: Crainic TG, Laporte G., editors. Fleet management and logistics. Dordrecht: Kluwer; 1998. p. 115–26.
[3] Powell WB, Jaillet P, Odoni A. Stochastic and dynamic networks and routing. In: Ball MO et al., editors. Network routing. Handbooks in operations research and management science, vol. 8, Amsterdam: Elsevier; 1995. p. 145–295.
[4] Psaraftis HN. Dynamic vehicle routing problems. In: Golden BL, Assad AA., editors. Vehicle routing: methods and studies. Amsterdam: North-Holland; 1988. p. 223–48.
[5] Psaraftis HN. Dynamic vehicle routing: status and prospects. Annals of Operations Research 1995;61:143–64.
[6] Ichoua S, Gendreau M, Potvin J-Y. Vehicle dispatching with time-dependent travel times. European Journal of Operational Research 2003;144:379–96.
[7] Fleischmann B, Gietz M, Gnutzmann S. Time varying travel times in vehicle routing. Transportation Science 2004;38: 160–73.
[8] Horn MET. Efficient modelling of travel in networks with time-varying link speeds. Networks 2000;36:80–90.
[9] Regan AC, Mahmassani HS, Jaillet P. Improving the efficiency of commercial vehicle operations using real-time information: potential uses and assignment strategies. Transportation Research Record 1995;1493:188–98.
[10] Taillard ED, Badeau P, Gendreau M, Guertin F, Potvin J-Y. A tabu search heuristic for the vehicle routing problem with soft time windows. Transportation Science 1997;31:170–86.