TRANSPORTATION RESEARCH PART B

Transportation Research Part B 38 (2004) 669–685

www.elsevier.com/locate/trb

Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows

Snežana Mitrović-Minić [a,*], Ramesh Krishnamurti [a], Gilbert Laporte [b]

[a] *School of Computing Science, Simon Fraser University, Burnaby, BC, Canada V5A 1S6*

[b] *Canada Research Chair in Distribution Management, HEC Montréal, 3000 Chemin de la Côte-Sainte-Catherine, Montreal, Canada H3T 2A7*

Received 10 March 2002; received in revised form 15 March 2003; accepted 11 September 2003

Abstract

The dynamic *Pickup and Delivery Problem with Time Windows* (PDPTW) is faced by courier companies serving same-day pickup and delivery requests for the transport of letters and small parcels. This article focuses on the dynamic PDPTW for which future requests are not stochastically modelled or predicted. The standard solution methodology for the dynamic PDPTW is the use of a rolling time horizon as proposed by Psaraftis. When assigning a new request to a vehicle it may be preferable to consider the impact of a decision both on a short-term and on a long-term horizon. In particular, better managing slack time in the distant future may help reduce routing cost. This paper describes double-horizon based heuristics for the dynamic PDPTW. Computational results show the advantage of using a double-horizon in conjunction with insertion and improvement heuristics.
© 2003 Elsevier Ltd. All rights reserved.

Keywords: Dynamic pickup and delivery problem with time windows; On-line heuristic; Rolling time horizon; Short-term and long-term time horizons

1. Introduction

The courier service industry has grown steadily over the last decades. Every large city in the world has several dozen courier companies often serving a thousand same-day pickup and delivery requests for the transport of letters and small parcels. These transportation requests arrive during the whole day, and usually only a small percentage of the requests are known in advance.

* Corresponding author. Tel.: +1-604-291-5938; fax: +1-604-291-3045.
E-mail address: smitrovi@cs.sfu.ca (S. Mitrović-Minić).

0191-2615/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/j.trb.2003.09.001

Assigning the requests to vehicles, routing the vehicles, and scheduling the routes, are operations typically performed manually by a dispatcher. Since efficient dispatching is critical to the operations of a courier company, there is a need for research on methods and strategies for solving the real-time dispatching of pickup and delivery vehicles.

With advances in telecommunication and information systems such as Global Positioning Systems (GPS), Geographical Information Systems (GIS), and Intelligent Transportation Systems (ITS), it has become realistic to solve dynamic routing problems in real-time and the importance of developing workable algorithms for such problems has increased.

1.1. Problem statement

Let $G = (V, A)$ be a graph where V is a set of locations and A is a set of arcs. The set V consists of a set of vehicle starting positions $D = \{s(v_1), s(v_2), s(v_3), \ldots\}$, a set of pickup locations $P^+ = \{1^+, 2^+, 3^+, \ldots, n^+\}$, and a set of delivery locations $P^- = \{1^-, 2^-, 3^-, \ldots, n^-\}$. A pair of locations (i^+, i^-) defines a transportation request r_i for picking up a load at i^+ and delivering it at i^-. With every arc $(i, j) \in A$, $i \neq j$, $i, j \in V$ are associated a distance $d_{i,j}$ and a travel time $t_{i,j}$. A fleet of vehicles serve the requests but its size is not fixed since for the most part the fleet is comprised of vehicles owned by drivers under contract. Each driver starts its route at his home base; there is therefore no central depot in this problem.

Each stop location i from the set $P = P^+ \cup P^-$ has to be served within a specified hard time window $[a_i, b_i]$, where a_i is the release time and b_i is the deadline. However, if a vehicle arrives too early at a given location, it is allowed to wait. The time windows are wide, often ranging from 30 min to several hours. The load of a transportation request comprises letters or small parcels, and is very small compared to vehicle capacity. Therefore, vehicle capacity is not binding. The problem *service period* is determined by the hours of operation of a courier company, and typically lasts about 10 h during a day. The *Pickup and Delivery Problem with Time Windows* (PDPTW) consists of determining a set of optimal routes for a fleet of vehicles in order to serve transportation requests. The objective is to minimize total route length, i.e., the sum of the distances traveled by all the vehicles, under the following constraints: all requests must be served, time windows must be respected, each request must be served entirely by one vehicle (*pairing constraint*), and each pickup location has to be served before its corresponding delivery location (*precedence constraint*).

If the pickup and delivery locations of every request coincide, the PDPTW reduces to the *Multiple Traveling Salesman Problem with Time Windows*. The PDPTW is NP-hard, and even deciding whether there exists a feasible solution when the number of vehicles is fixed, is NP-complete in the strong sense (Savelsbergh, 1985).

The *dynamic* PDPTW arises when not all requests are known in advance. A difficulty in solving such a problem is illustrated in Fig. 1. The hollow circles are delivery locations from requests previously assigned to vehicles A and B. Fig. 1a shows the best routes if request r_8 (with pickup location 8^+ and delivery location 8^-) becomes known at time t', and request r_9 (with locations 9^+ and 9^-) becomes known at time $t'' > t'$. Request r_8 is inserted in route A at time t', making the future insertion of request r_9 in the same route infeasible. At time t'', the reassignment of request r_8 is not allowed due to pairing constraint (vehicle A has already visited 8^+). Therefore, the total distance traveled is larger than the distance needed to travel if all requests had been known in advance (Fig. 1b).

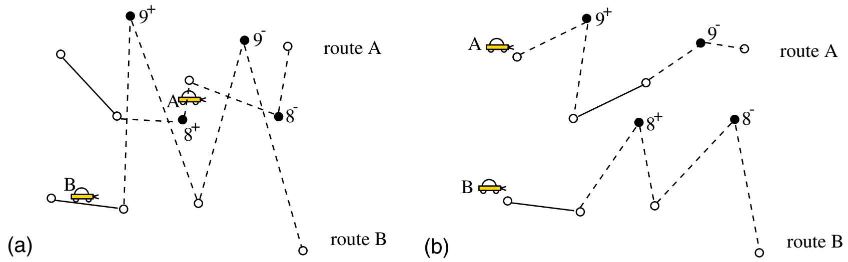

Fig. 1. The dynamic pickup and delivery problem with time windows. (a) The best feasible insertion of requests r_8 and r_9 in the dynamic environment. The time is t''. (b) The best feasible insertion of requests r_8 and r_9 if they were known at time t' or earlier.

A feasible solution to the dynamic PDPTW consists of a set of open routes and a set of schedules. A route is a sequence of stop locations assigned to one vehicle. Each route begins at the vehicle's starting position, and ends at a delivery location. A route schedule is comprised of a list of pairs—the scheduled arrival and departure times of a location—for each location in the route. The article concentrates on solving routing and scheduling problems arising in real-time, using on-line algorithms. Each on-line routing solution has to be followed by an on-line scheduling of the route, because in dynamic problems vehicles usually have to serve some of the assigned requests before new requests become known.

1.2. Literature review

There is an abundant body of research on the PDPTW. We distinguish between *Full-Truckload* dispatching problems (see, e.g., Yang et al., 2000; Powell et al., 1995, 2000a,b; Regan et al., 1996a,b, 1998), and *Less-than-Truckload* (LTL) problems. The applications covered by this article belong to the second category. The majority of studies on the LTL pickup and delivery problems deal with capacity constrained problems (PDPTWC). Among these, the most studied problems are related to passenger transportation: the *Dial-a-Ride Problem* (DARP) and the *Handicapped Transportation Problem* (HTP). These are characterized by tight capacity constraints and narrow time windows (typically ranging from 10 to 30 min). In addition, the maximal passenger riding time can be twice the direct riding time, and a vehicle is not allowed to wait while carrying a passenger. Surveys on solution methods can be found in Bodin et al. (1983), Assad (1988), Desrosiers et al. (1995) and Cordeau and Laporte (2003). Reports specifically focused on dynamic vehicle routing problems can be found in Psaraftis (1988, 1995), and Gendreau and Potvin (1998).

1.2.1. The PDPTW with capacity constraints

Optimization algorithms for the PDPTWC include dynamic programming methods and the Dantzig-Wolfe decomposition (column generation) method. Dynamic programming has been applied to the single-vehicle DARP. Instances with ten (Psaraftis, 1983) and 40 requests (Desrosiers et al., 1986) have been solved. The column generation method (Dumas et al., 1991) was able to solve the multiple-vehicle version of the problem as well. The PDPTWC instances solved were characterized by 55 requests, 22 vehicles, and loads ranging from 30% to 100% of the vehicle capacity.

Heuristics developed for the PDPTWC may be classified as decomposition methods, construction methods, improvement methods, and incomplete optimization heuristics (using mathematical programming methods). The most successful of these has been an incomplete optimization method based on column generation. The method consists of first clustering requests, and then designing optimal routes. It has been applied to large-scale instances: 880 requests for the DARP (Desrosiers et al., 1988; Dumas et al., 1989), and more than 2400 requests for the HTP (Desrosiers et al., 1991; Ioachim et al., 1995). Unfortunately, the approach is very sensitive to the tightness of the time windows and capacity constraints and tends not to perform well when these constraints are loose. When applied to the traveling salesman problem with time windows, the method "experiences extreme degeneracy difficulties" (Dumas et al., 1995, p. 367). We also mention two other articles of interest in this area. The first, by Daganzo (1978), develops an approximate model for many-to-many transportation systems without time windows. The second, by Fu (2002), concentrates on the approximation of stochastic travel times in the PDPTWC.

Modern heuristics were recently applied to the PDPTWC. Thus, Toth and Vigo (1997) have proposed a tabu threshold procedure for solving a real-world HTP with 312 requests. Potvin and Rousseau (1992) have developed a constraint-directed search algorithm for the DARP, and have tested it on the instances of Jaw et al. (1986) containing 10 requests per hour, over a 9 h period. A reactive tabu search approach has also been proposed (Nanry and Barnes, 2000) and has been tested on modified Solomon's instances of the vehicle routing problem with time windows with 100 requests. More recently, Cordeau and Laporte (2003) have developed a tabu search heuristic capable of solving static real-life DARP instances containing up to 295 requests. The quality of solutions produced by modern heuristics is strongly related to running time. If sufficient time is given, the algorithms attain near optimal or even optimal solutions, as borne out by empirical studies. However, the time available for decision making in a real-time dynamic environment is often short and a different approach is needed in such contexts.

1.2.2. The PDPTW without capacity constraints

Little work has been done on the PDPTW without capacity constraints. There are no reports on exact algorithms for this problem, and only a few heuristics have been developed. One of the early heuristics combines a variable-depth arc-exchange procedure with simulated annealing (Van der Bruggen et al., 1993). The method reached near-optimal solutions on the static single-vehicle PDPTW instances with 38 requests, and with time windows that were small compared to the average travel time. Shen et al. (1995) have developed an expert system using a neural network as a learning module. The system was tested on real-life instances with 140 requests, 12 vehicles, and a 6-h service period. Thirty minutes were allowed for a pickup, and 90 min for a delivery, from the time a request came in. Another decision support system with learning capabilities uses a utility function pre-constructed by a genetic programming technique, aimed at approximating the decisions of a dispatcher (Benyahia and Potvin, 1998). The real-life instances of Shen et al. (1995) were used for testing. Gendreau et al. (1998a) have solved the dynamic pickup and delivery problem with soft time windows using a tabu search procedure. Ejection chains were used to define neighborhoods, and an adaptive memory (Rochat and Taillard, 1995)—a repository of routes associated with the best visited solutions—was used to diversify the search, while a decomposition technique was implemented to intensify the search. The method was tested on instances with 24 requests per hour, over the 6-h service period, and on instances with 33 requests per hour, over the 4-h service period. The authors also reported on a parallel method implementation.

Available algorithms for the PDPTWC are not always applicable to the PDPTW without capacity constraints since they exploit vehicle capacity (for example, capacity constraints help limit the cardinality of the search space in dynamic programming and column generation methods). In addition some algorithms will fail when applied to large-scale instances. Powell (1998) suggests that when a dynamic routing problem becomes more complicated, good solution strategies must become simpler. This is supported by the fact that despite their relative simplicity, many insertion, construction and local search heuristics have produced good solutions in complex practical environments. We aim to use these findings, and combine them with characteristics of the dynamic PDPTW to arrive at an efficient solution procedure.

One difficulty when solving PDPTWs is to make good short-term decisions without adverse long-term effects. For example using a myopic policy in the short-term may remove the flexibility needed to make good long-term decisions. We therefore propose the use of a *double-horizon* when assigning a new request to a vehicle and when scheduling the vehicle. Our results show that this approach is superior to the standard *rolling horizon* (Psaraftis, 1988) often used in PDPTW algorithms.

The remainder of this article is organized as follows. Section 2 provides a formal description of a dynamic optimization problems, introduces the concept of short-term and long-term goals, and defines double-horizon based heuristics. Section 3 describes empirical study and provides computational results. Conclusions follow in Section 4.

2. Heuristics for the dynamic PDPTW

Before proceeding to the description of our double-horizon based heuristics for the dynamic PDPTW, we introduce some general concepts on dynamic combinatorial optimization problems and on short-term and long-term goals.

2.1. Dynamic combinatorial optimization problems

A dynamic combinatorial optimization problem arises when data are gradually revealed over a service period. Each dynamic problem $\mathscr{I}$ has its corresponding static problem I. Denote by F an objective function for $\mathscr{I}$ and by f an objective function for I, i.e., a function that would have been used had all the data been available. The aim is to solve $\mathscr{I}$ so that at the end of the service period the solution achieved is as close as possible to an optimal solution x^* of I with respect to f. The function F need not be equal to f, but f still provides guidance in determining F.

Instance $\mathscr{I}$ may be seen as a set of static instances I_t, each of which is a snapshot of $\mathscr{I}$ at time t. An optimal solution x_t^* of the static instance I_t with respect to F is called a *tentative solution*. A solution actually implemented during the time interval $[t_i, t_{i+1})$ is called a *permanent solution* and is labeled $\dot{x}_{[t_i,t_{i+1})}$ (Fig. 2).

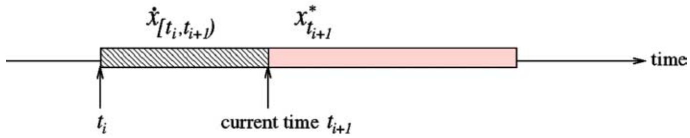

Fig. 2. Let $x^*_{t_i}$ and $x^*_{t_{i+1}}$ be two tentative solutions when solving the dynamic optimization problem. The permanent solution $\dot{x}_{[t_i,t_{i+1})}$ is equal to the $[t_i, t_{i+1})$ portion of $x^*_{t_i}$ if no tentative solution is found between t_i and t_{i+1}.

Suppose that tentative solutions $x^*_0, x^*_\delta, x^*_{2\delta}, x^*_{3\delta}, \ldots,$ are determined at time instants $0, \delta, 2\delta, 3\delta, \ldots$ A solution $\dot{x}(F, \delta)$ to $\mathscr{I}$ is a sequence of permanent solutions $(\dot{x}_{[0,\delta)}, \dot{x}_{[\delta,2\delta)}, \dot{x}_{[2\delta,3\delta)}, \ldots)$. It depends on both F and δ. The value of an optimal solution $\dot{x}^*$ to instance $\mathscr{I}$ is

$$\min_F \left\{ \lim_{\delta \to 0} f(\dot{x}(F, \delta)) \right\}$$

for a minimization problem. The function F^* yielding $\dot{x}^*$ may be distinguished as an *optimal objective function* corresponding to f.

2.2. Short-term and long-term goals

When solving the dynamic PDPTW with wide time windows a route defined by the tentative solution x^*_t may be bound for the next several hours: the pickup location may be scheduled for service in the near future, while the corresponding delivery location may be scheduled for service only much later. Thus, x^*_t may easily span a significant portion of the service period.

Assume that an optimal solution x^* to I is known in advance. At time t determine a tentative solution x^*_t to I_t, and compare the $[t_i, t_{i+1})$ portion of x^*_t to that of x^*, where $t_i \geqslant t$. When the time interval $[t_i, t_{i+1})$ is in the near future compared to t, it is likely that most locations belonging to the $[t_i, t_{i+1})$ portion of x^* are already known, even though some new requests may appear between t and t_i. In contrast, when the time interval $[t_i, t_{i+1})$ is in the distant future, many requests belonging to the $[t_i, t_{i+1})$ portion of x^* may not be known at time t. It is therefore imperative to maintain a special structure for the $[t_i, t_{i+1})$ portion of the tentative solution x^*_t in order to allow effective future insertions. For example, if we want to minimize the total distance traveled, it may be preferable to accumulate slack time in the distant future rather than concentrating on distance minimization, since larger slack times make future request insertions easier.

It would therefore seem advisable to differentiate between a *short-term goal* applying to the first portion of the tentative solution x^*_t, and a *long-term goal* applying to the portion of x^*_t in the distant future. The short-term goal is to reduce traveled distance, while the long-term goal is to maintain the routes in a state that will enable them to easily respond to future requests. The short-term goal is similar to the objective function of the corresponding static optimization problem, while the long-term goal need not be similar.

2.3. Double-horizon based heuristics

A *double-horizon* based heuristic solves a dynamic problem with a short-term and a long-term goal (Fig. 3). It differs from the rolling horizon heuristic introduced by Psaraftis (1988) which

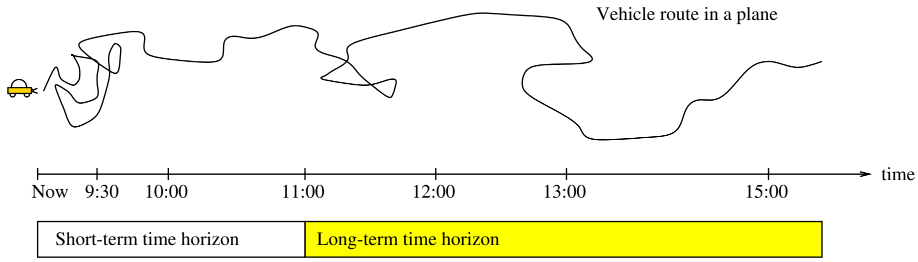

Fig. 3. A vehicle route in a plane, the time when the vehicle will drive along specific parts of the route, and the two time horizons.

operates with a dynamically redefined short-term horizon. As mentioned at the end of Section 1.1, the solution of a dynamic PDPTW contains a routing and a scheduling component. This article proposes double-horizon based heuristics for vehicle routing and scheduling.

2.3.1. The routing subproblem

We have developed two routing heuristics. The first is a constructive heuristic consisting of a cheapest insertion procedure and a cheapest reinsertion procedure. The second heuristic is similar but contains an improvement procedure (Fig. 4).

The cheapest insertion procedure is applied to new requests accumulated over a certain time period of length δ. It is followed by the reinsertion of all scheduled requests whose pickup location has not yet been served. Before insertion these requests are sorted in increasing order of slack

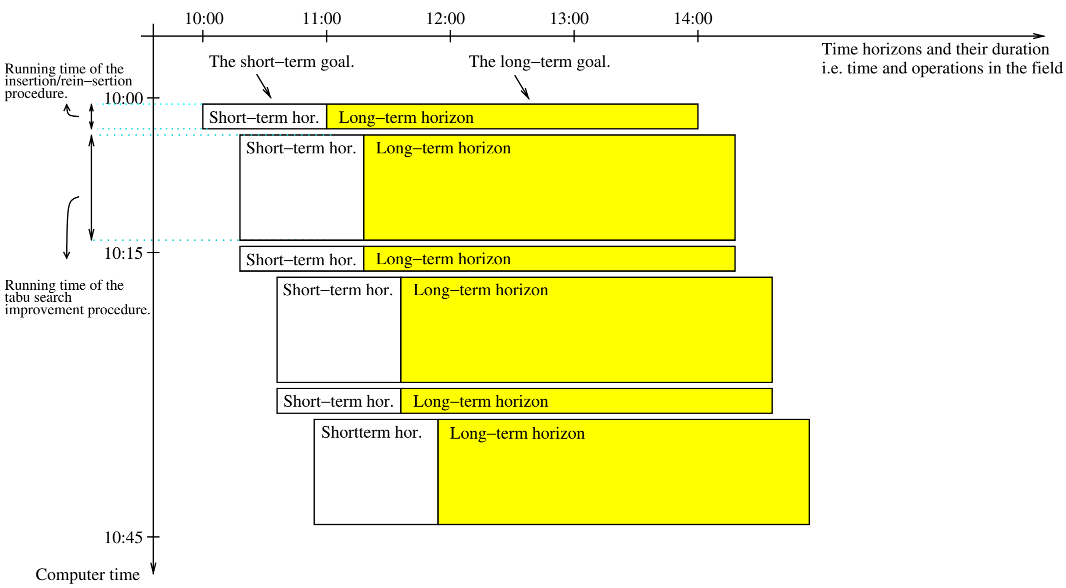

Fig. 4. The double-horizon based heuristic applied on the routing subproblem of the dynamic PDPTW. The solution method consists of the insertion/reinsertion procedure and the tabu search improvement procedure. The length of the short-term horizon is 1 h, and the improvement procedure runs for 15 min.

time. The slack time of a request is equal to the difference between the total time available to serve the request and the direct travel time between its pickup and delivery locations. The reinsertions are performed using the same cheapest insertion procedure.

The improvement procedure is based on tabu search. It is applied after the reinsertion procedure and it runs while new requests are being accumulated. The tabu search procedure is a simplified version of the method introduced by Gendreau et al. (1998a), with neighborhoods defined by means of ejection chains. We use a greedy approach to extend the current ejection chain. Solution feasibility is maintained at all times because of the difficulty experienced by some researchers (Garcia et al., 1994; Gendreau et al., 1998b) in regaining feasibility when hard time windows are violated.

The two goals used in the double-horizon heuristics we have developed for the routing subproblem are (1) the minimization of the total route length over the short-term time horizon, and (2) the minimization of a linear combination of distance and time over the long-term time horizon. These two goals are implemented by using an insertion cost c and an objective function z.

The insertion cost is used in the constructive heuristic. It is defined as

$$c = [(1-\alpha_p)f_p + \alpha_p g_p] + [(1-\alpha_d)f_d + \alpha_d g_d],$$

where f_p and f_d are the route length increases due to the insertion of a pickup and delivery locations, and where g_p and g_d are the corresponding decreases in slack times. We have considered three variants of the insertion cost:

(1) c_1: increase in route length obtained by setting $\alpha_p = \alpha_d = 0$;
(2) c_2: convex combination of route length increase and slack time decrease with $0 < \alpha < 1$;
(3) c_3: different costs over two time horizons by setting $\alpha_p = \alpha_d = 0$ if the pickup and delivery location falls within the short-term horizon and using $0 < \alpha < 1$ otherwise.

Note that only the third case corresponds to a double-horizon based heuristic. The first two were used for comparison purposes.

The objective function z is used in the tabu search procedure and has the following form:

$$z = \frac{l}{s}\beta q_S + (1-\beta)((1-\alpha)q_L + \alpha h_L),$$

where q_S is total length of the route portions belonging to the short-term time horizon, q_L is the total length of remaining parts of the routes, h_L is the sum over all routes of the average slack times over all locations in the long-term time horizon, l is the length of the long-term time horizon, and s is the length of the short-term time horizon. The values of α and β are determined empirically.

We have considered three variants of the objective function:

(1) z_1: total route length obtained by setting $\alpha = \beta = 0$ and where q_L is interpreted as the total route length;
(2) z_2: convex combination of route length and slack time obtained by setting $0 < \alpha < 1$ and $\beta = 0$ and where q_L is interpreted as the total route length and h_L is interpreted as the sum over all routes of the average slack times;

(3) z_3: different values of the objective function over two time horizons are obtained by setting $0 < \alpha < 1$ and $0 < \beta < 1$.

Again only the third case corresponds to a double-horizon based heuristic.

In the routing heuristic containing the cheapest insertion procedure and the tabu search improvement procedure, the following combinations of insertion cost and objective function are explored: (c_1, z_1), (c_2, z_2), and (c_3, z_3).

2.3.2. The scheduling subproblem

Once a partial vehicle route is defined, the scheduling subproblem consists of determining the departure time of each vehicle from each yet unvisited location of its planned route. Several strategies are available at this stage. In a dynamic environment, it is common to schedule a route by applying a *Drive-First* (DF) strategy in which a vehicle drives as soon as possible, and waits only if it reaches a location before its release time. This is so mainly because this strategy is the only appropriate one in solving a static vehicle routing problem.

In a double-horizon based heuristic better scheduling may be achieved by suitably planning the waiting periods in the later portion of the routes. The available waiting time in a route is organized into a few large waiting intervals which are arranged along the whole route. The route is partitioned into segments, each containing consecutive locations that are reasonably close to each other in the plane (note that a route may contain locations that are far apart due to the pairing constraints). Segments change dynamically as new locations are inserted in a route or removed from it. A segment may split in two, or two segments may be merged. Such decisions are made using a prespecified maximal area for the rectangle enclosing all locations of a segment.

Each vehicle drives as soon as possible, serving all locations of the same segment. When the last location is served, the vehicle waits a certain period of time before heading to the first location of the next segment. For good results this waiting period is only a portion of the maximum possible waiting time, i.e., the vehicle will leave the current location at a time instant between the earliest possible departure time and the latest possible departure time. The time interval that the vehicle spends serving locations in the first segment corresponds to the short-term time horizon. The remainder of the route belongs to the long-term time horizon. This strategy, called the *Advanced Dynamic Waiting* (ADW) strategy, was selected after extensive computational tests. For a full description and comparative analysis of several scheduling strategies, see Mitrović-Minić (2001) and Mitrović-Minić and Laporte (in press).

3. Empirical study and computational results

We have carried out several experiments with our heuristics for the dynamic PDPTW. Evaluating the quality of a heuristic for a dynamic problem where neither the optimal solution nor strong lower bounds is known—as is the case in the dynamic PDPTW—is not an easy task. Two possible ways are to establish comparisons with solutions achieved by another heuristic, or with solutions obtained for the corresponding static instances. We have used both types of comparison. In the first case, we have compared our approach with classical single-horizon based heuristics. In the second case, we have defined the *value of information* as a measure of effectiveness of a method for solving a dynamic optimization problem. Given an optimal solution $\dot{x}^*$ of the dynamic instance $\mathscr{I}$, and an optimal solution x^* of the static instance I with respect to the objective function f, we define the *value of information*

$$V = \frac{f(\dot{x}^*) - f(x^*)}{f(\dot{x}^*)},$$

which measures the possible gain in solving the dynamic problem if all information is known in advance. However, since we know neither of the two optimal solutions, we approximate V by $V(\mathscr{H})$, the *value of information under heuristic* $\mathscr{H}$, obtained when both the dynamic and the corresponding static problems, are solved by the same heuristic $\mathscr{H}$:

$$V(\mathscr{H}) = \frac{f(\dot{x}^{\mathscr{H}}) - f(x^{\mathscr{H}})}{f(\dot{x}^{\mathscr{H}})},$$

where $\dot{x}^{\mathscr{H}}$ is the solution to $\mathscr{I}$ obtained by heuristic $\mathscr{H}$, and $x^{\mathscr{H}}$ is the solution to I obtained by $\mathscr{H}$. Thus, $V(\mathscr{H})$ is a measure of the effectiveness of heuristic $\mathscr{H}$ for solving the dynamic problem, and we have used it in our empirical study.

3.1. Test instances

The experiments were performed on two sets of instances. Each set contains instances with 100, 500 and 1000 requests. There are 30 instances of each problem size in each set. The two sets differ in the distribution and width of time windows. The time window widths depend on the maximal time allowed to serve the whole request, assuming that the request is picked-up at the pickup release time. If this maximal time is k hours, the request is called a k-hour request. The requests were generated based on real-life data collected in two medium-to-large courier companies operating in Vancouver, Canada. In the first set of instances the distribution of requests is taken to be: 20% 1-h requests, 30% 2-h requests, and 50% 4-h requests. In the second set of instances the distribution of requests is: 10% 1-h requests, 20% 2-h requests, 30% 4-h requests, 30% 6-h requests, and 10% 8-h requests.

The service period is 10 h, the service area is 60×60 km^2, and vehicle speed is 60 km/h. Requests occur during the service period according to a continuous uniform distribution, and no request is known in advance. A request r_i is created by: (1) generating the time of its occurrence, (2) generating random positions of the pickup and delivery locations, and (3) generating k in order to define request r_i as a k-hour request. The time windows of the pickup location i^+ and the delivery location i^- are determined as follows: the pickup release time a_{i^+} is the time when the request occurs, the delivery deadline is $b_{i^-} = a_{i^+} + k$, the pickup deadline is $b_{i^+} = b_{i^-} - t_{i^+,i^-}$, and the start of the delivery time window is $a_{i^-} = a_{i^+} + t_{i^+,i^-}$. All requests can be feasibly served within the service period.

Rejections of requests and violations of time windows are not allowed. This is made possible by the fact that the fleet size is assumed to be unbounded. This is consistent with practice since a very large pool of private drivers can be used. Vehicle fixed costs are entirely borne by drivers. This is why we do not attempt to minimize the number of vehicles in our algorithms. The initial fleet is 20, 60, 80 for instances with 100, 500, 1000 requests, respectively. The same starting position at

(20 km, 30 km) was used for all vehicles since this scenario has shown to be fairer for comparisons among heuristics (Mitrović-Minić and Laporte, in press).

All experiments were carried out using a simulation speed in excess of one in order to solve many more instances of the dynamic problem in less time. A simulation speed s means that 1 h of real-life operations is simulated in $1/s$ hours of computer time. Also note that a simulation speed s used for solving a problem instance of size n indicates that even an instance of size $s \times n$ can be solved without any significant difficulties.

3.2. Assessment of the constructive routing heuristic

This section reports on the performance of the double-horizon based heuristics used for the routing and scheduling subproblems, when routing is solved by the constructive procedure.

The cheapest insertion procedure is applied every 15 min. It determines the overall best insertions for the locations of a request, with respect to c_1, c_2 and c_3 (see Section 2.3.1). Parameter α takes the value 0.25 which was determined after extensive preliminary testing. The short-term horizon length is one fourth of the total maximal time available for serving a request. More precisely, the short-term horizon length is 1 h in the first set of instances and 2 h in the second set. The remaining time corresponds to the long-term time horizon.

The scheduling problem is solved by means of the drive-first strategy (DF) and of the advanced dynamic waiting strategy (ADW), which is a double-horizon based heuristic. The short-term time horizon of a route corresponds to the first route segment. The remaining part of the route belongs to the long-term time horizon. The maximal area of the rectangle enclosing all the locations of a segment is 100 km^2.

We have solved the instances with 100, 500 and 1000 requests. The 100-request instances were tested with a simulation speed of 60, resulting in 1 h of real-time being simulated in 1 min of computer time. The 500-request instances and the 1000-request instances were tested with a simulation speed of 30, resulting in 1 h of real-time being simulated in 2 min of computer time.

Tables 1 and 2 provide comparisons of results averaged over 30 test instances. Boldface entries in columns 'c_3 vs. c_1' and 'c_3 vs. c_2' represent percentage improvements achieved by the double-horizon based heuristic used for routing with respect to two versions of a rolling horizon heuristic. They range from 0.64% to 5.60% with respect to total route length. Boldface entries in rows 'ADW vs. DF' show average improvements obtained by the double-horizon based heuristic used for scheduling, and they range from 1.38% to 8.53% in total route length. Improvements achieved when double-horizon based heuristics were used for both routing and scheduling (compared with combinations of (DF, c_1) and (DF, c_2)) appear in rectangles. They range from 4.05% to 10.36% in total route length. In all tables, one-tailed t-tests for paired comparisons were performed (see Kanji, 1993, p. 9). Significance levels are indicated by a star system: *** (0.01), ** (0.05), and * (0.10).

These results are consistent over all instances of Tables 1 and 2. The improvement in route length obtained when using c_3 is much superior when compared with c_2 than when compared with c_1. In fact using c_3 in conjunction with the ADW scheduling strategy is just marginally better than using a combination of ADW and c_1. The relative advantage of c_3 with respect to c_1 and c_2 tends to decrease as n increases. Irrespective of the routing strategy employed, the double-horizon scheduling heuristic (ADW) is always superior to DF but again the improvement is smaller for larger instances. Using a double-horizon based heuristic both for routing and scheduling (values

Table 1
Summary of computational results for the first set of instances (20% are 1-h requests, 30% are 2-h requests and 50% are 4-h requests)

n	Scheduling	Routing									
		c_1		c_2		c_3		c_3 **vs.** c_1 (%)		c_3 **vs.** c_2 (%)	
		Distance	m	Distance	m	Distance	m	Distance	m	Distance	m
100	DF	3021.48	19.63	3038.87	19.13	2929.74	19.13	**3.04*****	2.55	**3.59*****	0.00
	ADW	2789.50	19.23	2863.14	18.93	2745.98	18.97	**1.56*****	1.39	**4.09*****	−0.18
	ADW vs. DF (%)	**7.68*****	2.04	**5.78*****	1.05	**6.27*****	0.87	$\boxed{9.12^{***}}$	3.40	$\boxed{9.64^{***}}$	0.87
500	DF	10491.92	58.47	10643.41	55.17	10319.18	58.03	**1.65*****	0.74	**3.05*****	−5.20
	ADW	9936.64	55.70	10291.28	53.63	9816.63	55.37	**1.21*****	0.60	**4.61*****	−3.23
	ADW vs. DF (%)	**5.29*****	4.73	**3.31*****	2.78	**4.87*****	4.60	$\boxed{6.44^{***}}$	5.30	$\boxed{7.77^{***}}$	−0.36
1000	DF	18675.29	76.93	18888.27	77.07	18257.76	76.67	**2.24*****	0.35	**3.34*****	0.52
	ADW	17830.26	75.90	18279.55	74.30	17610.45	74.23	**1.23*****	2.20	**3.66*****	0.09
	ADW vs. DF (%)	**4.52*****	1.34	**3.22*****	3.59	**3.55*****	3.17	$\boxed{5.70^{***}}$	3.51	$\boxed{6.77^{***}}$	3.68

The table reports total distance traveled and number of vehicles used (m).

Table 2
Summary of computational results for the second set of instances (10% are 1-h requests, 20% are 2-h requests, 30% are 4-h requests, 30% are 6-h requests, and 10% are 8-h requests)

n	Scheduling	Routing									
		c_1		c_2		c_3		c_3 **vs.** c_1 (%)		c_3 **vs.** c_2 (%)	
		Distance	m	Distance	m	Distance	m	Distance	m	Distance	m
100	DF	2766.90	17.33	2793.40	16.13	2721.19	17.10	**1.65****	1.35	**2.58*****	−5.99
	ADW	2530.97	16.17	2652.50	16.20	2504.05	17.03	**1.06***	−5.36	**5.60*****	−5.14
	ADW vs. DF (%)	**8.53*****	6.73	**5.04*****	−0.41	**7.98*****	0.39	$\boxed{9.50^{***}}$	1.73	$\boxed{10.36^{***}}$	−5.58
500	DF	9692.29	50.07	9912.80	47.87	9541.86	50.33	**1.55*****	−0.53	**3.74*****	−5.15
	ADW	9167.70	48.00	9603.97	46.50	9107.89	48.33	**0.65**	−0.69	**5.17*****	−3.94
	ADW vs. DF (%)	**5.41*****	4.13	**3.12*****	2.86	**4.55*****	3.97	$\boxed{6.03^{***}}$	3.46	$\boxed{8.12^{***}}$	−0.97
1000	DF	16754.57	73.03	17070.84	72.63	16615.31	72.77	**0.83****	0.37	**2.67*****	−0.18
	ADW	16180.51	71.70	16834.58	71.13	16076.74	71.23	**0.64***	0.65	**4.50*****	−0.14
	ADW vs. DF (%)	**3.43*****	1.83	**1.38*****	2.07	**3.24*****	2.11	$\boxed{4.05^{***}}$	2.46	$\boxed{5.82^{***}}$	1.93

The table reports total distance traveled and number of vehicles used (m).

shown in rectangles) is always much better than using a rolling horizon heuristic (c_1 or c_2 for routing, DF for scheduling). Statistics relative to the number of vehicles used in the solution are highly variable and are only provided for information purposes since we did not attempt to minimize fleet size.

3.3. Assessment of the improvement procedure

This section reports on the performance of the double-horizon based heuristics used for routing and for scheduling, when a tabu search procedure is added to the constructive routing heuristic.

The tabu search procedure is applied every 15 min (when $s = 1$) and runs close to 15 min. It improves the current solution with respect to z_1, z_2 and z_3 (see Section 2.3.1). Parameter α takes the value 0.25 which is the same as in the insertion cost formula (Section 3.2), and parameter β is equal to 0.7. Again, these values were empirically determined. The short-term horizon length is one fourth of the total maximal time available for serving a request. The short-term horizon length is again 1 h in the first set of instances and 2 h in the second set of instances. The remaining time corresponds to the long-term time horizon. The cheapest insertion procedure is applied every 15 min and assigns to vehicles the new requests accumulated in the last 15 min. The insertion costs c_1, c_2 and c_3 were computed as in Section 3.2.

We have solved 100-request and 500-request instances. The 100-request instances were tested with a simulation speed of 10, resulting in the tabu search taking 1.2 min. The 500-request instances were tested with a simulation speed of 5, with the tabu search improvement procedure running for 2.4 min.

Tables 3 and 4 give comparisons of results averaged over 30 test instances. Boldface entries in columns 'c_3 vs. c_1' and 'c_3 vs. c_2' represent improvements achieved by the double-horizon based

Table 3
Summary of computational results for the first set of instances (20% are 1-h requests, 30% are 2-h requests and 50% are 4-h requests)

n	Scheduling	Routing									
		c_1		c_2		c_3		c_3 **vs.** c_1 (%)		c_3 **vs.** c_2 (%)	
		Distance	m	Distance	m	Distance	m	Distance	m	Distance	m
100	DF	2945.74	19.63	2981.03	18.43	2922.78	19.37	**0.78**	1.36	**1.95****	−5.06
	ADW	2742.38	19.17	2840.38	18.53	2755.70	19.03	**−0.49**	0.70	**2.98*****	−2.70
	ADW vs. DF (%)	**6.90*****	2.38	**4.72*****	−0.54	**5.72*****	1.72	$\boxed{6.45^{***}}$	3.06	$\boxed{7.56^{***}}$	−3.25
500	DF	10168.44	56.47	10447.51	53.47	10138.57	56.03	**0.29**	0.77	**2.96*****	−4.80
	ADW	9839.65	55.40	10161.74	52.23	9804.15	54.20	**0.36**	2.17	**3.52*****	−3.77
	ADW vs. DF (%)	**3.23*****	1.89	**2.74*****	2.31	**3.30*****	3.27	$\boxed{3.58^{***}}$	4.01	$\boxed{6.16^{***}}$	−1.37

The table reports total distance traveled and number of vehicles used (m).

Table 4
Summary of computational results for the second set of instances (10% are 1-h requests, 20% are 2-h requests, 30% are 4-h requests, 30% are 6-h requests, and 10% are 8-h requests)

n	Scheduling	Routing									
		c_1		c_2		c_3		c_3 **vs.** c_1 (%)		c_3 **vs.** c_2 (%)	
		Distance	m	Distance	m	Distance	m	Distance	m	Distance	m
100	DF	2733.85	17.47	2739.95	15.77	2708.39	16.73	**0.93**	4.20	**1.15***	−6.13
	ADW	2518.54	16.97	2594.16	15.47	2538.04	16.50	**−0.77**	2.75	**2.16*****	−6.68
	ADW vs. DF (%)	**7.88*****	2.86	**5.32*****	1.90	**6.29*****	1.39	$\boxed{7.16^{***}}$	5.53	$\boxed{7.37^{***}}$	−4.65
500	DF	9532.93	48.30	9734.66	46.57	9502.90	48.47	**0.32**	−0.35	**2.38*****	−4.08
	ADW	9104.13	47.83	9534.10	46.10	9159.50	48.20	**−0.61**	−0.77	**3.93*****	−4.56
	ADW vs. DF (%)	**4.50*****	0.97	**2.06*****	1.00	**3.61*****	0.55	$\boxed{3.92^{***}}$	0.21	$\boxed{5.91^{***}}$	−3.51

The table reports total distance traveled and number of vehicles used (m).

heuristic used for routing. They range from −0.77% to 3.93% with respect to total route length. Boldface entries in rows 'ADW vs. DF' show average improvements obtained by the double-horizon based heuristic used for scheduling, and they range from 2.06% to 7.88% in total route length. Improvements achieved when double-horizon based heuristics were used for both routing and scheduling, range from 3.58% to 7.56% (in total route length) compared to the results achieved when routing and scheduling were solved without the double-horizon. Regarding routing, improvements achieved by c_1 compared to c_3 are smaller (in terms of total route length) when the improvement procedure is used. In fact, in three out of eight cases c_3 is better than c_1, but only marginally. Overall, other comments made in Section 3.2 on the performance of the constructive routing heuristic apply to the improvement procedure.

3.4. The value of information

To determine the value of information, off-line experiments were performed on the corresponding static instances with 100 and 500 requests. In off-line experiments, routing is solved by means of the cheapest insertion procedure using c_1, followed by the tabu search improvement procedure using z_1. The tabu search procedure runs for 30 min. In on-line experiments, routing is solved by means of the constructive procedure using c_3, followed by the tabu search improvement procedure using z_3. The scheduling strategy does not affect the solution of a static instance. The off-line results are compared to the on-line results where scheduling is solved either by the DF waiting strategy or by the double-horizon based heuristic (ADW). Tables 5 and 6 give the values of information obtained for the two sets of instances.

Table 5
The value of information (multiplied by 100) for the heuristic in which routing is solved by the tabu search improvement procedure over the first set of instances

n	Scheduling	On-line		Off-line		Value of information × 100	
		Distance	m	Distance	m	Distance	m
100	DF	2922.78	19.37	2325.39	13.90	**20.44**% shorter routes	28.24% fewer vehicles
	ADW	2755.70	19.03	2325.39	13.90	**15.62**% shorter routes	26.96% fewer vehicles
500	DF	10138.57	56.03	8769.96	34.07	**13.50**% shorter routes	39.19% fewer vehicles
	ADW	9804.15	54.20	8769.96	34.07	**10.55**% shorter routes	37.14% fewer vehicles

Table 6
The value of information (multiplied by 100) for the heuristic in which routing is solved by the tabu search improvement procedure over the second set of instances

n	Scheduling	On-line		Off-line		Value of information × 100	
		Distance	m	Distance	m	Distance	m
100	DF	2708.39	16.73	2143.16	10.57	**20.87**% shorter routes	36.82% fewer vehicles
	ADW	2538.04	16.50	2143.16	10.57	**15.56**% shorter routes	35.94% fewer vehicles
500	DF	9502.90	48.47	8022.20	28.13	**15.58**% shorter routes	41.96% fewer vehicles
	ADW	9159.50	48.20	8022.20	28.13	**12.41**% shorter routes	41.64% fewer vehicles

The value of information is always positive as it may be expected. In the case of routing costs, it is smaller for larger instances, implying that less distance would be gained by knowing all requests in advance for the larger instances. The reverse relation is observed for the number of vehicles.

4. Conclusion

This article introduces the concept of double-horizon based heuristics for the dynamic pickup and delivery problem with time windows. Such heuristics may be applied to both routing and scheduling subproblems. They solve a problem over two time horizons using two goals. The idea is useful in contexts where problem solutions span a significant part of the service period, and in situations where what happens in the near future depends on the solution proposed for the distant future. Extensive computational experiments have shown that the use of a double-horizon can yield gain in route costs when compared with classical (single) rolling horizon methods, but percentage improvement tends to go down as instances become larger.

Acknowledgements

This research was partly supported by the Canadian Natural Sciences and Engineering Research Council under grants RGPIN36809 and OGP0039682. This support is gratefully acknowledged. Thanks are due to the referees for their valuable comments.

References

Assad, A.A., 1988. Modeling and implementation issues in vehicle routing. In: Golden, B.L., Assad, A.A. (Eds.), Vehicle Routing: Methods and Studies, Studies in Management Science and Systems, vol. 16. North-Holland, Amsterdam, pp. 7–45.

Benyahia, I., Potvin, J.-Y., 1998. Decision support for vehicle dispatching using genetic programming. IEEE Transactions on Systems, Man and Cybernetics—Part A: Systems and Humans 28, 306–314.

Bodin, L.D., Golden, B.L., Assad, A.A., Ball, M.O., 1983. Routing and scheduling of vehicles and crews: the state of the art. Computers & Operations Research 11, 63–211.

Cordeau, J.-F., Laporte, G., 2003. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. Transportation Research 37B, 579–594.

Daganzo, C.F., 1978. An approximate analytical model of many-to-many demand responsive transportation systems. Transportation Research 12, 325–333.

Desrosiers, J., Dumas, Y., Solomon, M.M., Soumis, F., 1995. Time constrained routing and scheduling. In: Ball, M.O., Magnanti, T.L., Monma, C.L., Nemhauser, G.L. (Eds.), Network Routing, Handbooks in Operations Research and Management Science, vol. 8. North-Holland, Amsterdam, pp. 35–139.

Desrosiers, J., Dumas, Y., Soumis, F., 1986. A dynamic programming solution of the large-scale single-vehicle dial-a-ride problem with time windows. American Journal of Mathematical and Management Sciences 6, 301–325.

Desrosiers, J., Dumas, Y., Soumis, F., 1988. The multiple vehicle dial-a-ride problem. In: Daduna, J.R., Wren, A. (Eds.), Computer Aided Transit Scheduling. Proceedings of the Fourth International Workshop on Computer-Aided Scheduling of Public Transport Lecture Notes in Economics and Mathematical Systems 308. Springer-Verlag, Berlin, pp. 15–27.

Desrosiers, J., Dumas, Y., Soumis, F., Taillefer, S., Villeneuve, D., 1991. An algorithm for mini-clustering in handicapped transport. Les cahiers du GERAD G-91-02, École des Hautes Études Commerciales, Montreal.

Dumas, Y., Desrosiers, J., Gélinas, E., Solomon, M.M., 1995. An optimal algorithm for the traveling salesman problem with time windows. Operations Research 43, 367–371.

Dumas, Y., Desrosiers, J., Soumis, F., 1989. Large-scale multi-vehicle dial-a-ride problems. Les cahiers du GERAD G-89-30, École des Hautes Études Commerciales, Montreal.

Dumas, Y., Desrosiers, J., Soumis, F., 1991. The pickup and delivery problem with time windows. European Journal of Operational Research 54, 7–22.

Fu, L., 2002. Scheduling dial-a-ride paratransit under time-varying, stochastic congestion. Transportation Research 36B, 485–506.

Garcia, B.-L., Potvin, J.-Y., Rousseau, J.-M., 1994. A parallel implementation of the tabu search heuristic for vehicle routing problems with time windows constraints. Computers & Operations Research 21, 1025–1033.

Gendreau, M., Guertin, F., Potvin, J.-Y., Séguin, R., 1998a. Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries. Report CRT-98-10, Centre de recherche sur les transports, Montreal.

Gendreau, M., Hertz, A., Laporte, G., Stan, M., 1998b. A generalized insertion heuristic for the traveling salesman problem with time windows. Operations Research 43, 330–335.

Gendreau, M., Potvin, J.-Y., 1998. Dynamic vehicle routing and dispatching. In: Crainic, T.G., Laporte, G. (Eds.), Fleet Management and Logistics. Kluwer, Boston, pp. 115–126.

Ioachim, I., Desrosiers, J., Dumas, Y., Solomon, M.M., Villeneuve, D., 1995. A request clustering algorithm for door-to-door handicapped transportation. Transportation Science 29, 63–78.

Jaw, J.-J., Odoni, A.R., Psaraftis, H.N., Wilson, N.H.M., 1986. A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. Transportation Research 20B, 243–257.

Kanji, G.K., 1993. 100 Statistical Tests. Sage, London.

Mitrović-Minić, S., 2001. The dynamic pickup and delivery problem with time windows. Ph.D. Dissertation, School of Computing Science, Simon Fraser University, Burnaby, Canada.

Mitrović-Minić, S., Laporte, G., in press. Waiting strategies for the dynamic pickup and delivery problem with time windows. Transportation Research B, in press.

Nanry, W.P., Barnes, J.W., 2000. Solving the pickup and delivery problem with time windows using reactive tabu search. Transportation Research 34B, 107–121.

Potvin, J.-Y., Rousseau, J.-M., 1992. Constraint-directed search for the advanced request dial-a-ride problem with service quality constraints. In: Balci, O., Shrada, R., Zenios, S.A. (Eds.), Computer Science and Operations Research: New Developments in Their Interfaces. Pergamon Press, Oxford, pp. 457–474.

Powell, W.B., 1998. On languages for dynamic resource scheduling problems. In: Crainic, T.G., Laporte, G. (Eds.), Fleet Management and Logistics. Kluwer, Boston, pp. 127–157.

Powell, W.B., Jaillet, P., Odoni, A.R., 1995. Stochastic and dynamic networks and routing. In: Ball, M.O., Magnanti, T.L., Monma, C.L., Nemhauser, G.L. (Eds.), Network Routing, Handbooks in Operations Research and Management Science, vol. 8. North-Holland, Amsterdam, pp. 141–295.

Powell, W.B., Snow, W., Cheung, R.K., 2000a. Adaptive labeling algorithms for the dynamic assignment problem. Transportation Science 34, 50–66.

Powell, W.B., Towns, M.T., Marar, A., 2000b. On the value of globally optimal solutions for dynamic routing and scheduling problems. Transportation Science 34, 67–85.

Psaraftis, H.N., 1983. An exact algorithm for the single vehicle many-to-many dial-a-ride problem with time windows. Transportation Science 17, 351–357.

Psaraftis, H.N., 1988. Dynamic vehicle routing problems. In: Golden, B.L., Assad, A.A. (Eds.), Vehicle Routing: Methods and Studies, Studies in Management Science and Systems, vol. 16. North-Holland, Amsterdam, pp. 223–248.

Psaraftis, H.N., 1995. Dynamic vehicle routing: status and prospects. Annals of Operations Research 61, 143–164.

Regan, A.C., Mahmassani, H.S., Jaillet, P., 1996a. Dynamic decision making for commercial fleet operations using real-time information. Transportation Research Record 1537, 91–97.

Regan, A.C., Mahmassani, H.S., Jaillet, P., 1996b. Dynamic dispatching strategies under real-time information for carrier fleet management. In: Lesort, J.B. (Ed.), Transportation and Traffic Theory. Pergamon, Oxford, pp. 737–756.

Regan, A.C., Mahmassani, H.S., Jaillet, P., 1998. Evaluation of dynamic fleet management systems: simulation framework. Transportation Research Record 1645, 176–184.

Rochat, Y., Taillard, É.D, 1995. Probabilistic diversification and intensification in local search for vehicle routing. Journal of Heuristics 1, 147–167.

Savelsbergh, M.W.P., 1985. Local search in routing problems with time windows. Annals of Operations Research 4, 285–305.

Shen, Y., Potvin, J.-Y., Rousseau, J.-M., Roy, S., 1995. A computer assistant for vehicle dispatching with learning capabilities. Annals of Operations Research 61, 189–211.

Toth, P., Vigo, D., 1997. Heuristic algorithms for the handicapped persons transportation problem. Transportation Science 31, 60–71.

Van der Bruggen, L.J.J., Lenstra, J.K., Schuur, P.C., 1993. Variable-depth search for the single-vehicle pickup and delivery problem with time windows. Transportation Science 27, 298–311.

Yang, J., Jaillet, P., Mahmassani, H.S., 2000. Real-time multi-vehicle truckload pick-up and delivery problems. Working paper, Department of Industrial and Manufacturing Engineering, New Jersey Institute of Technology, Newark, NJ.