4OR 1: 89–101 (2003) **Invited Survey**

DOI: 10.1007/s10288-002-0009-8

# The Dial-a-Ride Problem (DARP): Variants, modeling issues and algorithms

**Jean-François Cordeau, Gilbert Laporte**

GERAD-HEC Montréal, 3000 chemin de la Côte-Sainte-Catherine,
Montréal, Canada H3T 2A7 (e-mail: gilbert.laporte@hec.ca)

Received: June 2002 / Revised version: August 2002

**Abstract.** The *Dial-a-Ride Problem* (DARP) consists of designing vehicle routes and schedules for $n$ users who specify pick-up and drop-off requests between origins and destinations. The aim is to plan a set of $m$ minimum cost vehicle routes capable of accommodating as many users as possible, under a set of constraints. The most common example arises in door-to-door transportation for elderly or disabled people. The purpose of this article is to review the scientific literature on the DARP. The main features of the problem are described and classified and some modeling issues are discussed. A summary of the most important algorithms is provided.

**Keywords:** dial-a-ride problem, survey, static and dynamic pick-up and delivery problems

**AMS classification:** 90B06, 90C27, 90C59

## 1 Introduction

The *Dial-a-Ride Problem* (DARP) consists of designing vehicle routes and schedules for $n$ users who specify pick-up and drop-off requests between origins and destinations. Very often the same user will have two requests during the same day: an *outbound* request from home to a destination (e.g., a hospital), and an *inbound* request for the return trip. In the standard version, transport is supplied by a fleet of $m$ identical vehicles based at the same depot. The aim is to plan a set of minimum cost vehicle routes capable of accommodating as many requests as possible, under a set of constraints. The most common example arises in door-to-door transportation services for elderly or disabled people (Madsen et al. 1995; Toth and Vigo 1996, 1997; Borndörfer et al. 1997).

**4OR** Quarterly Journal of the Belgian, French and Italian Operations Research Societies

In western countries several local authorities are setting up dial-a-ride services or are overhauling existing systems in response to increasing demand. This phenomenon can be attributed in part to the ageing of the population but also to a trend toward the development of ambulatory health care services. Some existing systems cannot adequately meet demand while others are faced with escalating operating costs. There is a genuine need for reliable cost effective systems, and operational research can help reach this goal.

From a modeling point of view, the DARP generalizes a number of vehicle routing problems such as the *Pick-up and Delivery Vehicle Routing Problem* (PDVRP) and the *Vehicle Routing Problem with Time Windows* (VRPTW). For overviews on these problems, see Desrosiers et al. (1995) and Desaulniers et al. (2002). What makes the DARP different from most such routing problems is the human perspective. When transporting passengers, reducing user inconvenience must be balanced against minimizing operating costs. In addition, vehicle capacity is normally constraining in the DARP whereas it is often redundant in PDVRP applications, particularly those related to the collection and delivery of letters and small parcels.

The purpose of this article is to review the scientific literature specific to the DARP. It is organized as follows. In Sect. 2 the main features of the DARP are described and classified, and some modeling issues are discussed. A summary of the most important algorithms is provided in Sect. 3, followed by conclusions in Sect. 4.

## 2 Main features of the DARP

Dial-a-ride services may operate according to a *static* or to a *dynamic* mode. In the first case, all transportation requests are known beforehand while in the second case requests are gradually revealed throughout the day and vehicle routes are adjusted in real-time to meet demand. In practice pure dynamic DARPs rarely exist since a subset of requests is often known in advance.

Most studies on the DARP assume the availability of a fleet of $m$ homogeneous vehicles based at a single depot. While this hypothesis often reflects reality and can serve as a sound base for the design of models and algorithms, it is important to realize that different situations exist in practice. There may be several depots, especially in wide geographical areas, and the fleet is sometimes heterogeneous. Some vehicles are designed to carry wheelchairs only, others may only cater to ambulatory passengers and some are capable of accommodating both types of passenger. The main consideration in some problems is to first determine a fleet size and composition capable of satisfying all demand while in other contexts, the aim is to maximize the number of requests that can be served with a fixed size fleet. Some systems routinely turn down several requests each day. A compromise consists of serving some of the demand with a core vehicle fleet and using extra vehicles (e.g., taxis) if necessary.

Given this, it makes sense to consider two possible problems: 1) minimize costs subject to full demand satisfaction and side constraints; 2) maximize satisfied demand subject to vehicle availability and side constraints. The most common cost elements relate to regular fleet size and operation, occasional use of extra vehicles, and driver wages.

Quality of service criteria include route duration, route length, customer waiting time, customer ride time (i.e., total time spent in vehicles), and difference between actual and desired drop-off times. Some of these criteria may be treated as constraints or as part of the objective function. A common trend in DARP models is to let users impose a time window on both their departure and arrival times. We believe this may be unduly constraining for the transporter, particularly if these time windows are narrow. Following Jaw et al. (1986) we believe that users should be able to specify a time window on the arrival time of their outbound trip and on the departure time of their inbound trip. The transporter then determines a planned departure time for the outbound trip and a planned arrival time for the inbound trip, while satisfying an upper bound on the ride time. In practice, since travel times are somewhat uncertain, the outbound departure time communicated to the user should be slightly earlier than the scheduled time.

## 3 Algorithms

Three important decisions are associated with the construction of a DARP solution: 1) determining *clusters* of users to be served by the same vehicle; 2) *sequencing* these users into a vehicle route; 3) *scheduling* pick-up, driving and drop-off activities along each route. Some algorithms execute these steps sequentially while others take a more holistic perspective and intertwine these decisions. We will first present the scheduling aspect which plays an important role in several DARP algorithms. This will be followed by a description of the best known algorithms under two headings: single-vehicle DARP and multi-vehicle DARP.

### *3.1 Scheduling*

Given a route $k = (v_0, \dots, v_i, \dots, v_q)$ consisting of a sequence of vertices, where $v_0$ and $v_q$ both represent the depot, the scheduling problem is to determine the departure time from the depot and the time at which service should begin at each vertex $v_1, \dots, v_{q-1}$, so that time windows are satisfied and route duration is minimized. This problem is of critical importance whenever an upper bound is imposed on route duration.

We use the following notation:

$T_k$ : the maximal duration of route $k$;
$[e_i, \ell_i]$ : a time window on the beginning of service at vertex $v_i$ (every vehicle must leave the depot no earlier than $e_0$ and return no later than $\ell_0$);
$t_{ij}$ : the travel time from $v_i$ to $v_j$;
$d_i$ : the service duration at $v_i$;
$A_i$ : the arrival time of the vehicle at $v_i$;
$B_i$ : the time at which service begins at $v_i$;
$D_i$ : the departure time from $v_i$;
$W_i$ : the waiting time at $v_i$.

Note that $B_i \geq \max\{e_i, A_i\}$ and $D_i = B_i + d_i$. The time window at $v_i$ is violated if $B_i > \ell_i$. Arrival at $v_i$ before $e_i$ is allowed and therefore the waiting time at that vertex is $W_i = B_i - A_i$.

If the scheduling problem is feasible, a solution can be identified by sequentially setting $B_0 = e_0$ and $B_i = \max\{e_i, A_i\}$ for $i = 1, \ldots, q$. To reduce route duration and unnecessary waiting time, it may be advantageous to delay departure from the depot and the beginning of service at pick-up vertices. For this, one must compute for each $v_i$, the maximum delay $F_i$ that can be incurred before service starts so that no time window in route $k$ will be violated. Savelsbergh (1992) calls $F_i$ the *forward time slack* of $v_i$. It is computed as

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

which can be rewritten as

$$F_i = \min_{i \leq j \leq q} \left\{ \sum_{i < p \leq j} W_p + (\ell_j - B_j) \right\}, \tag{2}$$

since $B_j = B_i + \sum_{i \leq p < j} (t_{p,p+1} + d_p) + \sum_{i < p \leq j} W_p$.

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 $v_j$. The optimal departure time from the depot can thus be determined in $O(q)$ time by computing $F_0$. Whenever the vehicle becomes empty, minimizing the ride time of the first user to be picked-up can be achieved by computing the forward time slack $F_i$ of the corresponding vertex $v_i$.

Instead of route duration, some authors have minimized an *inconvenience function* $f_k$ computed in terms of the $B_i$ variables. Sexton and Bodin (1985a) consider the case where $f_k = \sum_{i=1}^{q-1} \alpha_i B_i$ and the $\alpha_i$'s are preset parameters. If $v_{i+1}$ denotes the successor of $v_i$ in route $k$, the problem can be formulated more generally as an optimization problem of the form

$$\text{Minimize} \quad \sum_{i=0}^{q} g_i(B_i) \tag{3}$$

subject to

$$B_i - B_{i+1} \leq -t_{i,i+1} - d_i \qquad (i = 0, \ldots, q-1) \tag{4}$$

$$-B_i \leq -e_i \qquad (i = 0, \ldots, q) \tag{5}$$

$$B_i \leq \ell_i \qquad (i = 0, \ldots, q), \tag{6}$$

where $g_i(B_i)$ is a convex function defined with respect to the time window $[e_i, l_i]$. Dumas et al. (1989) have proposed a dual approach to solve this problem by performing $q$ unidimensional minimizations. In the special cases where the inconvenience functions are quadratic or linear, the complexity of the algorithm is $O(q)$.

Finally, Hunsacker and Savelsbergh (2002) have devised a procedure for efficiently testing the feasibility of an insertion in construction or improvement heuristics. They consider a variant of the DARP with time windows, an upper bound on $W_i$, and an upper bound on the ride time, proportional to the driving time. They have shown how to check in $O(q)$ time whether the insertion of a given request in a route is feasible.

### 3.2 The single-vehicle DARP

One of the simplest cases of the DARP is where all requests are known in advance and all users are served by a single-vehicle. Psaraftis (1980) formulated and solved the problem as a dynamic program in which the objective function is the minimization of the weighted sum of route completion time and customer dissatisfaction. Customer dissatisfaction is itself expressed as a weighted combination of waiting time before pick-up and ride time. Time windows are not specified by users. Instead the transporter imposes “maximum position shift” constraints limiting the difference between the position of a user in the calling list and its position in the vehicle route. This algorithm was later updated by the same author (Psaraftis, 1983) to handle user-specified time windows on departure and arrival times. As is often the case in dynamic programming formulations, the algorithm can only solve relatively small instances optimally since the procedure has an $O(n^2 3^n)$ complexity. The largest instance solved using this approach contains nine users. While most DARPs arising in practice are much larger, the proposed approach could still prove useful as a subroutine in a multi-vehicle algorithm, provided the number of users in each route remains relatively small.

Sexton (1979) and Sexton and Bodin (1985a, 1985b) also view the single-vehicle DARP as a step in a multi-vehicle DARP heuristic in which the users have previously been clustered. Their algorithm iterates between solving a routing problem by means of an insertion heuristic and solving the associated scheduling problem. They formally describe the alternation of these two steps in the context of Benders decomposition. These authors minimize a user inconvenience function made up of the weighted sum of two terms. The first measures the difference between the actual travel time and the direct travel time of a user. The second term is the (positive) difference between desired drop-off time and actual drop-off time, under the assumption that the former is at least as large as the latter, late drop-offs being disallowed. As explained in Sect. 3.1, this objective can be expressed as a linear function of the $B_i$ variables. Results are reported on several data sets from Baltimore and Gaithersburgh, where the number of users varies between 7 and 20.

The single-vehicle DARP was reformulated as an integer program by Desrosiers et al. (1986). The formulation includes time windows as well as vehicle capacity and precedence constraints and it is solved exactly by dynamic programming. Using a double labelling scheme, the authors were capable of identifying and later eliminating several dominated states and state transitions. Optimal solutions were obtained for $n = 40$.

The dynamic single-vehicle DARP was also considered by Psaraftis (1980). In this problem, new requests occur dynamically in time but no information on future requests is available (unlike what happens in stochastic programming). When a new request becomes known at time $t$ a planned solution is available. All requests scheduled before $t$ have already been processed and are no longer relevant. The problem is then to reoptimize the portion of the solution from time $t$, including the new request. This is done by applying the dynamic programming algorithm developed for the static case. One practical difficulty stemming from this approach is being capable of solving the problem at time $t$ before the arrival of the next request, which may not be feasible if the algorithm is slow and requests arrive in quick succession. One way around this difficulty, recently proposed by Gendreau et al. (2001) in the context of dynamic ambulance relocation, is to precompute several scenarios, using parallel computing, in anticipation of future requests. Despite its limitations, Psaraftis's work on the dynamic single-vehicle DARP has helped define the concepts used in later research on dynamic routing problem (see Psaraftis, 1988; Psaraftis 1995; Mitrović-Minić et al. 2002).

### 3.3 The multi-vehicle DARP

One of the first heuristics for the multiple-vehicle static DARP was proposed by Jaw et al. (1986). The model considered by these authors imposes windows on the pick-up times of inbound requests and on the drop-off times of outbound requests. A maximum ride time, expressed as a linear function of the direct ride time, is imposed for each user. In addition, vehicles are not allowed to be idle when carrying passengers. A non-linear objective function combining several types of disutility is used to assess the quality of solutions. The heuristic selects users in order of earliest feasible pick-up time and gradually inserts them into vehicle routes so as to yield the least possible increase of the objective function. The algorithm was tested on artificial instances involving 250 users and on a real data set with 2617 users and 28 vehicles.

A commonly used technique in such problems consists of defining clusters of users to be served by the same vehicle, prior to the routing phase. This idea is exploited by Bodin and Sexton (1986) who construct the clusters by grouping users who are close together in a combined space and time dimension before applying to each cluster the single-vehicle algorithm of Sexton and Bodin (1985a,b) and making swaps between the clusters. Results are presented on two instances extracted from a Baltimore data base and containing approximately 85 users each. Dumas et al. (1989) later improved upon this two-phase approach by creating so-called "mini-clusters" of users, i.e., groups of users to be served within the same area at approximately the same time. These mini-clusters are then optimally combined to form feasible vehicle routes, using a column generation technique. Finally, each vehicle route is reoptimized by means of the single-vehicle algorithm of Desrosiers et al. (1986), and a scheduling step is executed. The authors have successfully solved instances derived from real-life data taken from three Canadian cities: Montreal, Sherbrooke and Toronto. Instances with up to 200 users are easily solved, while larger instances require the use of a spatial and temporal decomposition technique. The mini-clustering phase was later improved by Desrosiers et al. (1991) who presented results on a data set comprising almost 3000 users. Finally, Ioachim at al. (1995) showed there was an advantage, in terms of solution quality, to resorting to an optimization technique to construct the clusters.

A real-life problem arising in Bologna was tackled by Toth and Vigo (1996). Users specify requests with a time window on their origin or destination. A limit proportional to direct distance is imposed on the ride time. Transportation is supplied by a fleet of capacitated minibuses and special cars. On occasions, taxis can be used but since these are not the best mode of transportation for disabled people, a penalty is imposed on their use. The objective is to minimize the total cost of service. Toth and Vigo have developed a heuristic consisting of first assigning requests to routes by means of a parallel insertion procedure, and then performing intra-route and inter-route exchanges. Tests performed on instances involving between 276 and 312 requests show significant improvements with respect to the previous hand-made solutions. Further improvements were later obtained (Toth and Vigo, 1997) through the execution of a tabu thresholding post-optimization phase after the parallel insertion step.

Another study, by Borndörfer et al. (1997), also uses a two-phase approach in which clusters of users are first constructed and then grouped together to form feasible vehicle routes. A cluster is defined as a "maximal subtour such that the vehicle is never empty". Its two end-points correspond to the pick-up of the first user and the drop-off of the last user, respectively. In the first phase, a large set of good clusters is constructed and a set partitioning problem is then solved to select a subset of clusters serving each user exactly once. In the second phase, feasible routes are enumerated by combining clusters and a second set partitioning problem is solved to select the best set of routes covering each cluster exactly once. Both set partitioning problems are solved by a branch-and-cut algorithm. On real-life instances, the algorithm cannot always be run to completion so that it must stop prematurely with the best known solution. It was applied to instances including between 859 and 1771 transportation requests per day in Berlin.

Wolfler Calvo and Colorni (2002) have devised a heuristic for a version of the DARP in which the number of available vehicles is fixed and windows are imposed on pick-up and drop-off times. A hierarchical objective function is used: the algorithm first attempts to service as many users as possible and then minimizes user inconvenience expressed as the sum of waiting time and excess ride time. The heuristic first constructs a set of $m$ routes and a number of subtours by solving an assignment problem. A routing phase is then performed to insert the subtours in the $m$ routes and to resequence the vertices within the routes. Tests were carried out on instances involving between 10 and 180 users.

The latest heuristic on the multi-vehicle static DARP is due to Cordeau and Laporte (2002). It applies tabu search to the following problem. Users specify a window on the arrival time of their outbound trip and on the departure time of their inbound trip, and a maximum ride time is associated with each user. It can either be the same for all users, or computed by using a maximum deviation factor from the most direct ride time of each particular user. Capacity and maximum route length constraints are imposed on the vehicles. The search algorithm iteratively removes a transportation request and reinserts it into another route. As is now commonly done in such contexts (Gendreau et al. 1994; Cordeau et al. 2001), intermediate infeasible solutions are allowed through the use of a penalized objective function. Also, the minimum duration schedule associated with each candidate solution is computed, as explained in Sect. 3.1. The algorithm was tested on randomly generated instances ($24 \leq n \leq 144$) and on six data sets ($n = 200$ and 295) provided by a Danish transporter. With respect to alternative algorithms such as column generation and branch-and-cut, tabu search can easily accommodate a large variety of constraints and objectives, even if these are non-linear.

Relatively little research on the multi-vehicle dynamic DARP is reported in the scientific literature. One interesting case is described by Madsen et al. (1995) who have solved a real-life problem involving services to elderly and disabled people in Copenhagen. Users may specify a desired pick-up or drop-off time window, but not both. Vehicles of several types are used to provide service, not all of which are available at all times. Requests arrive dynamically throughout the day, vehicle speeds are variable and vehicles may become unavailable due to breakdowns. The authors have developed an insertion algorithm, called REBUS, based on the procedure previously developed by Jaw et al. (1986). New requests are dynamically inserted in vehicle routes taking into account their difficulty of insertion into an existing route. The algorithm was tested on a 300-customer, 24-vehicle problem. The

**Table 1.** Comparison of several DARP algorithms

| **Reference** | **Type** | **Objective** | **Time windows** | **Other constraints** | **Algorithm** | **Size of problems solved** |
|---|---|---|---|---|---|---|
| Psaraftis (1980) | Single-vehicle, static and dynamic. | Minimize a combination of route duration, ride time and waiting time. | None. | Vehicle capacity. Maximum position shift. | Exact. Dynamic programming. | $n \leq 9$. |
| Psaraftis (1983) | Single-vehicle, static. | Minimize route duration. | On pick-up and drop-off. | Vehicle capacity. Maximum position shift. | Exact. Dynamic programming. | $n \leq 9$. |
| Sexton (1979), Sexton and Bodin (1985a,b) | Single-vehicle, static. | Minimize weighted sum of differences between actual and desired drop-off times, and differences between actual and shortest possible ride times. | Upper bounds on pick-up and drop-off times. | Vehicle capacity. | Heuristic. Iterates between routing and scheduling phases. | $7 \leq n \leq 20$. |
| Desrosiers et al. (1986) | Single-vehicle, static. | Minimize route duration. | On pick-up or drop-off. | Vehicle capacity. | Exact. Dynamic programming. | $n \leq 40$. |
| Jaw et al. (1986) | Multi-vehicle, static. | Minimize non-linear combination of several types of disutility. | On pick-up or drop-off. | Vehicle capacity. Actual ride time cannot exceed a given percentage of minimum ride time. | Heuristic. Insertions. | $n = 250$ and $n = 2617$. |
| Bodin and Sexton (1986) | Multi-vehicle, static. | Minimize weighted sum of differences between actual and desired drop-off times, and differences between actual and shortest possible ride times. | Upper bounds on pick-up and drop-off times. | Vehicle capacity. | Heuristic. Iterates between routing and scheduling phases. | $n \approx 85$. |

**Table 1.** (continued)

| Reference | Type | Objective | Time windows | Other constraints | Algorithm | Size of problems solved |
|---|---|---|---|---|---|---|
| Dumas et al. (1989), Desrosiers et al. (1991), Ioachim et al. (1995) | Multi-vehicle, static. | Minimize number of vehicles used, then minimize total route duration. | On pick-up and drop-off. | Several vehicle types. Vehicle capacity. Maximum route duration. | Heuristic. Create mini-clusters. Group them by column generation. Apply scheduling phase. | $n \leq 1890$ in DDS (1989), $n = 2411$ in DDSTV (1991), $n = 2545$ in IDDS (1995) |
| Toth and Vigo (1996, 1997) | Multi-vehicle, static. | Minimize total service cost. | On pick-up or drop-off. | Vehicle capacity. Maximum ride time. | Heuristic. Parallel insertions followed by intra-route and inter-route exchanges in TV (1996) and also tabu thresholding in TV (1997). | $276 \leq n \leq 312$. |
| Börndorfer et al. (1997) | Multi-vehicle, static. | Minimize operational costs (drivers and vehicles). | On pick-up and drop-off. | Several vehicle types. Vehicle capacity. Maximum route duration. | Heuristic. Set partitioning formulation solved by truncated branch-and-cut algorithm. | $859 \leq n \leq 1771$. |
| Wolfler Calvo and Colorni (2002) | Multi-vehicle, static. | Maximize number of users that can be served, then minimize user inconvenience. | On pick-up and drop-off. | Vehicle capacity. | Heuristic. Clusters constructed by assignment heuristic, followed by vertex reinsertions. | $10 \leq n \leq 180$. |
| Cordeau and Laporte (2002) | Multi-vehicle, static. | Minimize total route length. | On pick-up or drop-off. | Vehicle capacity. Maximum route duration. Maximum ride time. | Heuristic. Tabu search with vertex reinsertions. | $24 \leq n \leq 295$. |
| Madsen et al. (1995) | Multi-vehicle, dynamic. | Multi-criteria objective. | On pick-up or drop-off. | Several vehicle types. Vehicle capacity. Maximum route duration. Maximum deviation between actual and shortest possible ride times. | Heuristic. Vertex insertions. | $n = 300$. |

authors report that the algorithm was capable of generating good quality solutions within very short computing times.

As noted by Borndörfer et al. (1997), the distinction between static and dynamic DARPs is often blurred in practice since requests are often cancelled and, as a result, transporters may allow the introduction of new requests in a solution designed for a static problem. Also, as mentioned, dynamic DARPs rarely exist in a pure form since a number of requests are often known when planning starts. The difficulty is then to design seed vehicle routes for these requests with sufficient slack to accommodate future dynamic demand.

We present in Table 1 a summary of the algorithms just described.

## 4 Conclusion

The DARP is an important and difficult routing problem encountered in several contexts and likely to gain in importance in coming years. It shares several features with pick-up and delivery problems arising in courier services, but since it is concerned with the transportation of people, level of service criteria become more important. Thus punctuality, reduction of idle time and route directness are more critical in the DARP. After more than twenty years of research, it is fair to say that excellent heuristics exist for the static case. It is now possible to solve instances with several hundreds of users within reasonable times and it should be possible to apply decomposition techniques for larger instances involving, say, two or three thousand users. We believe more emphasis should now be put on the dynamic version of the problem. This involves the construction of an initial solution for a limited set of requests known in advance and the design of features capable of determining whether a new request should be served or not and if so, how existing routes should be modified to accommodate it. In the same spirit, it should be possible to update a partially built solution to deal with cancellations and other unforeseen events such as traffic delays and vehicle breakdowns. In this spirit, recent studies on the determination of dynamic shortest paths (Pallottino and Scutellà, 1998) and of stochastic congestion (Fu 2002) bear particular significance. Finally, advanced systems should make full use of new technologies such as vehicle positioning systems now common in the area of emergency medical services (Brotcorne et al. 2002).

*Acknowledgements.* This work was partly supported by a Strategic research grant provided by HEC Montréal, by the Quebec Government FCAR research program under grant 2002-GR-73080, and by the Canadian Natural Sciences and Engineering Research Council under grants 227837-00 and OGPO039682. This support is gratefully acknowledged. Thanks are also due to a referee who provided valuable comments.

## References

Bodin LD, Sexton T (1986) The multi-vehicle subscriber dial-a-ride problem. *TIMS Studies in Management Science* 2: 73–86

Borndörfer R, Klostermeier F, Grötschel M, Küttner C (1997) Telebus Berlin: Vehicle scheduling in a dial-a-ride system. *Technical Report SC 97–23*, Konrad-Zuse-Zentrum für Informationstechnik Berlin

Brotcorne L, Laporte G, Semet F (2002) Ambulance location and relocation models. *European Journal of Operational Research* (forthcoming)

Cordeau J-F, Laporte G, Mercier A (2001) A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society* 52: 928–936

Cordeau J-F, Laporte G (2002) A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research B* (forthcoming)

Desaulniers G, Desrosiers J, Erdmann A, Solomon MM, Soumis F (2002) *VRP with pickup and delivery.* In: Toth P, Vigo D (eds) The vehicle routing problem, SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, pp 225–242

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, 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, Montréal

Desrosiers J, Dumas Y, Solomon MM, Soumis F (1995) *Time constrained routing and scheduling.* In: Ball MO, Magnanti TL, Monma CL, Nemhauser GL (eds) Network Routing, Handbooks in Operations Research and Management Science, Elsevier Science, Amsterdam, vol 8, pp 35–139

Dumas Y, Desrosiers J, Soumis F (1989a) Large scale multi-vehicle dial-a-ride problems. *Les Cahiers du GERAD*, G–89–30, École des Hautes Études Commerciales, Montréal

Dumas Y, Soumis F, Desrosiers J (1989b) Optimizing the schedule for a fixed vehicle path with convex inconvenience cost. *Les Cahiers du GERAD, G–89–08*, École des Hautes Études Commerciales, Montréal

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

Gendreau M, Hertz A, Laporte G (1994) A tabu search heuristic for the vehicle routing problem. *Management Science* 40: 1276–1290

Gendreau M, Laporte G, Semet F (2001) A dynamic model and parallel tabu search algorithm for real-time ambulance relocation. *Parallel Computing* 27: 1641–1653

Hunsaker B, Savelsbergh MWP (2002) Efficient feasibility testing for dial-a-ride problems. *Operations Research Letters* 30: 169–173

Ioachim I, Desrosiers J, Dumas Y, Solomon MM (1995) A request clustering algorithm for door-to-door handicapped transportation. *Transportation Science* 29: 63–78

Jaw J, Odoni AR, Psaraftis HN, Wilson NHM (1986) A heuristic algorithm for the multi-vehicle advance-request dial-a-ride problem with time windows. *Transportation Research B* 20: 243–257

Madsen OBG, Ravn HF, Rygaard JM (1995) A heuristic algorithm for the a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. *Annals of Operations Research* 60: 193–208

Mitrović-Minić S, Krishnamurti R, Laporte G (2002) The double-horizon heuristic for the dynamic pickup and delivery problem with time windows (submitted for publication)

Pallottino P, Scutellà MG (1998) *Shortest path algorithms in transportation models: Classical and innovative aspects.* In: Marcotte P, Nguyen S (eds) Equilibrium and advanced transportation modelling, Kluwer, Boston, pp 245–281

Psaraftis HN (1980) A dynamic programming approach to the single-vehicle, many-to-many immediate request dial-a-ride problem. *Transportation Science* 14: 130–154

Psaraftis HN (1983) An exact algorithm for the single-vehicle many-to-many dial-a-ride problem with time windows. *Transportation Science* 17: 351–357

Psaraftis HN (1988) *Dynamic vehicle routing problems.* In: Golden BL, Assad AA (eds) Vehicle Routing: Method and Studies, North-Holland, Amsterdam, pp 223–248

Psaraftis HN (1995) Dynamic vehicle routing: Status and prospects. *Annals of Operations Research* 61: 143–164

Savelsbergh MWP (1992) The vehicle routing problem with time windows: Minimizing route duration. *ORSA Journal on Computing* 4: 146–154

Sexton T (1979) *The single vehicle many-to-many routing and scheduling problem.* Ph.D. dissertation, SUNY at Stony Brook

Sexton T, Bodin LD (1985a) Optimizing single vehicle many-to-many operations with desired delivery times: I. Scheduling. *Transportation Science* 19: 378–410

Sexton T, Bodin LD (1985b) Optimizing single vehicle many-to-many operations with desired delivery times: II. Routing. *Transportation Science* 19: 411–435

Toth P, Vigo D (1996) *Fast local search algorithms for the handicapped persons transportation problem.* In: Osman IH, Kelly JP (eds) Meta-heuristics: Theory and applications. Kluwer, Boston, pp 677–690

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

Wolfler Calvo R, Colorni A (2002) An approximation algorithm for the dial-a-ride problem (submitted for publication)