Ann Oper Res (2007) 153: 29–46
DOI 10.1007/s10479-007-0170-8

The dial-a-ride problem: models and algorithms

Jean-François Cordeau · Gilbert Laporte

Published online: 5 May 2007

Abstract The *Dial-a-Ride Problem* (DARP) consists of designing vehicle routes and schedules for n users who specify pickup and delivery 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 a summary of the most important models and algorithms is provided.

Keywords Dial-a-ride problem · Survey · Static and dynamic pickup and delivery problems

1 Introduction

The *Dial-a-Ride Problem* (DARP) consists of designing vehicle routes and schedules for n users who specify pickup and delivery 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

This is an updated version of a paper that appeared in *4OR* 1:89–101, 2003.

J.-F. Cordeau (✉)
Canada Research Chair in Logistics and Transportation and GERAD, HEC Montréal, 3000, chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7
e-mail: cordeau@crt.umontreal.ca

G. Laporte
Canada Research Chair in Distribution Management and GERAD, HEC Montréal, 3000, chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7
e-mail: gilbert@crt.umontreal.ca

door-to-door transportation services for elderly or disabled people (see, e.g., Madsen et al. 1995; Toth and Vigo 1996, 1997; Borndörfer et al. 1997; Colorni and Righini 2001; Diana and Dessouky 2004; Rekiek et al. 2006; Melachrinoudis et al. 2007).

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 aging 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 operations research can help reach this goal.

From a modeling point of view, the DARP generalizes a number of vehicle routing problems such as the *Pickup and Delivery Vehicle Routing Problem* (PDVRP) and the *Vehicle Routing Problem with Time Windows* (VRPTW). For overviews on these problems, see Cordeau et al. (2007a, 2007b). 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.

This paper updates and extends a previous survey published in *4OR* (Cordeau and Laporte 2003b). It is organized as follows. The main features of the DARP are described in Sect. 2, followed by mathematical models in Sect. 3. The following two sections summarize the main algorithms for the single-vehicle DARP and for the multi-vehicle DARP, respectively. Conclusions follow in Sect. 6.

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 drivers' 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 delivery 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, but 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 Mathematical models

Several models have been suggested for a number of variants of the DARP. These can be found in the papers reviewed in Sects. 4 and 5. We present in Sect. 3.1 the three-index formulation of Cordeau (2006) which is based on the modeling choices presented in Sect. 2. This is followed in Sect. 3.2 by a more compact two-index formulation due to Ropke et al. (2007). In addition, we will describe in Sect. 3.3 models for the so-called scheduling problem, arising as a subproblem in some heuristics which first determine a routing sequence and then attempt to reduce route durations.

3.1 A three-index formulation for the DARP

Cordeau (2006) formulates the DARP on a directed graph $G = (V, A)$. The vertex set V is partitioned into $\{\{0, 2n+1\}, P, D\}$ where 0 and $2n+1$ are two copies of the depot, $P = \{1, \ldots, n\}$ is the set of pickup vertices and $D = \{n+1, \ldots, 2n\}$ is the set of delivery vertices. A request is a couple $(i, n+i)$, where $i \in P$ and $n+i \in D$. To each vertex $v_i \in V$ are associated a load q_i, with $q_0 = q_{2n+1} = 0$, $q_i \geq 0$ for $i = 1, \ldots, n$ and $q_i = -q_{i-n}$ for $i = n+1, \ldots, 2n$, and a service duration $d_i \geq 0$ with $d_0 = d_{2n+1} = 0$. The arc set is defined as $A = \{(i, j)\colon i = 0,\ j \in P$, or $i, j \in P \cup D,\ i \neq j$ and $i \neq n+j$, or $i \in D,\ j = 2n+1\}$. The capacity of vehicle k is Q_k and the maximal duration of route $k \in K$ is denoted by T_k. The cost of traversing arc (i, j) with vehicle k is equal to c_{ij}^k, and the travel time of arc (i, j) is denoted by t_{ij}. The maximal ride time is denoted by L and the time window of vertex i is $[e_i, \ell_i]$.

The model uses binary three-index variables x_{ij}^k equal to 1 if and only if arc (i, j) is traversed by vehicle $k \in K$. In addition, let u_i^k be the time at which vehicle k starts servicing vertex i, w_i^k the load of vehicle k upon leaving vertex i, and r_i^k the ride time of user i (corresponding to request $(i, n+i)$ on vehicle k). The model is then as follows.

(DARP)

$$\text{Minimize} \quad \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} c_{ij}^k x_{ij}^k \tag{1}$$

subject to

$$\sum_{k \in K} \sum_{j \in V} x_{ij}^k = 1 \qquad (i \in P), \tag{2}$$

$$\sum_{i \in V} x_{0i}^k = \sum_{i \in V} x_{i,2n+1}^k = 1 \quad (k \in K), \tag{3}$$

$$\sum_{j\in V} x_{ij}^k - \sum_{j\in V} x_{n+i,j}^k = 0 \qquad (i \in P, k \in K), \tag{4}$$

$$\sum_{j\in V} x_{ji}^k - \sum_{j\in V} x_{ij}^k = 0 \qquad (i \in P \cup D, k \in K), \tag{5}$$

$$u_j^k \geq (u_i^k + d_i + t_{ij}) x_{ij}^k \qquad (i, j \in V, k \in K), \tag{6}$$

$$w_j^k \geq (w_i^k + q_j) x_{ij}^k \qquad (i, j \in V, k \in K), \tag{7}$$

$$r_i^k \geq u_{n+i}^k - (u_i^k + d_i) \qquad (i \in P, k \in K), \tag{8}$$

$$u_{2n+1}^k - u_0^k \leq T_k \qquad (k \in K), \tag{9}$$

$$e_i \leq u_i^k \leq \ell_i \qquad (i \in V, k \in K), \tag{10}$$

$$t_{i,n+i} \leq r_i^k \leq L \qquad (i \in P, k \in K), \tag{11}$$

$$\max\{0, q_i\} \leq w_i^k \leq \min\{Q_k, Q_k + q_i\} \qquad (i \in V, k \in K), \tag{12}$$

$$x_{ij}^k = 0 \text{ or } 1 \qquad (i, j \in V, k \in K). \tag{13}$$

In this formulation, constraints (2) and (4) ensure that each request is served once by the same vehicle, while constraints (3) and (5) guarantee that each vehicle starts and ends its route at the depot. Constraints (6) to (8) define starts of service times, vehicle loads and user ride times, respectively, while constraints (9) to (12) ensure that these will be feasible.

3.2 A two-index formulation for the DARP

More recently, Ropke et al. (2007) have proposed two models and a branch-and-cut algorithm for the PDP with time windows (PDPTW) and for the DARP, where all vehicles are identical. The PDPTW is a DARP without the maximum ride time constraints. Here we describe the better of the two models. It works with a homogeneous fleet of vehicles of capacity Q and two-index variables x_{ij}. In this model, R denotes a path, $\mathcal{R}$ is the set of infeasible paths with respect to time windows and maximum ride time constraints, $A(R)$ is the arc set of R, and $\mathcal{U}$ is the collection of all sets $S \subset P \cup D$ satisfying $3 \leq |S| \leq |V| - 2$, $0 \in S$, $2n+1 \notin S$ and there exists $i \in P \backslash S$ with $n+i \in S$. Let also $q(S) = \sum_{i\in S} q_i$ denote the total load of vertices in S. The model is as follows.

(PDPTW-DARP)

$$\text{Minimize} \quad \sum_{i\in V}\sum_{j\in V} c_{ij} x_{ij} \tag{14}$$

subject to

$$\sum_{i\in V} x_{ij} = 1 \qquad (j \in P \cup D), \tag{15}$$

$$\sum_{j\in V} x_{ij} = 1 \qquad (i \in P \cup D), \tag{16}$$

$$\sum_{i,j\in S} x_{ij} \leq |S| - 2 \qquad (S \in \mathcal{U}), \tag{17}$$

$$\sum_{i,j\in S} x_{ij} \leq |S| - \max\{1, \lceil |q(S)|/Q \rceil\} \qquad (S \subseteq P \cup D,\ |S| \geq 2), \tag{18}$$

$$\sum_{(i,j)\in A(R)} x_{ij} \le |A(R)| - 1 \quad (R \in \mathcal{R}), \tag{19}$$

$$x_{ij} = 0 \text{ or } 1 \quad (i, j \in V). \tag{20}$$

In this model, constraints (17) guarantee that pairing and precedence constraints will be satisfied, i.e., a pickup vertex i will appear on the same route and before its associated delivery vertex $n+i$, constraints (18) are capacity constraints, and constraints (19) eliminate infeasible paths.

3.3 The scheduling problem

Given a route $k = (v_0 = 0, \ldots, v_i, \ldots, v_q = 2n + 1)$ consisting of a sequence of $q + 1$ vertices of V, the scheduling problem is to determine the departure time from the depot and the time at which service should begin at each vertex $v_1, \ldots, v_{q-1}$, 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:

- $[e_i, \ell_i]$: 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 ℓ_{2n+1});
- 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;
- u_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 $u_i \ge \max\{e_i, A_i\}$ and $D_i = u_i + d_i$. The time window at v_i is violated if $u_i > \ell_i$. Arrival at v_i before e_i is allowed and therefore the waiting time at that vertex is $W_i = u_i - A_i$.

If the scheduling problem is feasible, a solution can be identified by sequentially setting $u_0 = e_0$ and $u_i = \max\{e_i, A_i\}$ 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 pickup 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 follows:

$$F_i = \min_{i\le j\le q} \left\{ \ell_j - \left(u_i + \sum_{i\le p<j} (t_{p,p+1} + d_p) \right) \right\}, \tag{21}$$

which can be rewritten as

$$F_i = \min_{i\le j\le q} \left\{ \sum_{i<p\le j} W_p + (\ell_j - u_j) \right\}, \tag{22}$$

since $u_j = u_i + \sum_{i\le p<j}(t_{p,p+1} + d_p) + \sum_{i<p\le j} W_p$.

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 F_0 from the depot can thus be determined in $O(q)$. 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 u_i variables. Sexton and Bodin (1985a) consider the case where $f_k = \sum_{i=1}^{q-1} \alpha_i u_i$ and the α_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

$$\text{Minimize} \quad \sum_{i=0}^{q} g_i(u_i) \tag{23}$$

subject to

$$u_i - u_{i+1} \le -t_{i,i+1} - d_i \quad (i = 0, \dots, q-1), \tag{24}$$

$$-u_i \le -e_i \quad (i = 0, \dots, q), \tag{25}$$

$$u_i \le \ell_i \quad (i = 0, \dots, q) \tag{26}$$

where $g_i(u_i)$ is a convex function defined with respect to the time window $[e_i, l_i]$. Dumas et al. (1989b) 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, we mention that Hunsaker 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.

4 The single-vehicle DARP

One of the simplest cases of the DARP is that where all users are served by a single vehicle. We will successively review algorithms for the static case, where all requests are known in advance, and for the dynamic case, where they are gradually revealed in real-time.

4.1 The static case

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 pickup 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 delivery time and actual delivery time, under the assumption that the former is at least as large as the latter, late deliveries being disallowed. As explained in Sect. 3.3, this objective can be expressed as a linear function of the u_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 $n = 40$.

4.2 The dynamic case

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 problems (see, e.g., Psaraftis 1988, 1995; Mitrović-Minić et al. 2004). The algorithms for the single-vehicle case are summarized in Table 1.

5 The multi-vehicle DARP

Considerably more research has been carried out on the multi-vehicle DARP. Our survey is again subdivided into the static case and the dynamic case.

5.1 The static case

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 pickup times of inbound requests and on the delivery 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 pickup time and gradually inserts them into vehicle routes so as to yield the least possible increase of the objective function. The algorithm was

Table 1 Summary of several algorithms for the static single-vehicle DARP

Reference	Objective	Time windows	Other constraints	Algorithm	Size of instances solved
Psaraftis (1980)	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)	Minimize route duration	On pickup and delivery	Vehicle capacity Maximum position shift	Exact. Dynamic programming	$n \leq 9$
Sexton (1979), Sexton and Bodin (1985a, 1985b)	Minimize weighted sum of differences between actual and desired delivery times, and differences between actual and shortest possible ride times.	Upper bounds on pickup and delivery times	Vehicle capacity	Heuristic. Iterates between routing and scheduling phases	$7 \leq n \leq 20$
Desrosiers et al. (1986)	Minimize route duration	On pickup and delivery	Vehicle capacity	Exact. Dynamic programming	$n \leq 40$

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 clusters before applying to each cluster the single-vehicle algorithm of Sexton and Bodin (1985a, 1985b) 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. (1989a) 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 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 et 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 and 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 pickup of the first user and to the delivery 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.

The Cordeau and Laporte (2003a) heuristic for the multi-vehicle static DARP 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 request. It can either be the same for all requests, or computed by using a maximum deviation factor from the most direct ride time of each particular trip. 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. 3.3. 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 nonlinear.

The paper by Aldaihani and Dessouky (2003) considers a hybrid DARP in which two types of service are offered: door-to-door transportation, and transportation to and from a bus stop. The choice between the two modes is made within the algorithm. The authors have developed a two-part heuristic consisting of an insertion-reinsertion descent phase, followed by tabu search. Two objectives are considered: minimization of vehicle traveled distance and minimization of passenger traveled time. Either objective is considered in different parts of the algorithms. The authors report results using data from Lancaster County in California. Diana and Dessouky (2004) have applied a parallel regret based insertion procedure to solve a version of the DARP in which the objective is a weighted sum of distance, excess ride time over direct travel time, and vehicle idle time. They have reported results on instances of sizes 500 and 1000.

In 2006, four heuristics were published on the multi-vehicle DARP. In the paper by Rekiek et al. (2006), the main objective is the minimization of the number of vehicles used.

The authors propose a genetic algorithm for the clustering phase and an insertion mechanism for the routing phase. They report good results on data provided by the City of Brussels ($100 \leq n \leq 164$). Xiang et al. (2006) solve an elaborate version of the DARP in which the objective is the minimization of a combination of vehicle fixed costs, vehicle variable costs, driver costs, waiting time, and service time under several operating constraints. Insertions and inter-route exchanges are used to construct the routes. The authors introduce an element of diversification in their search mechanism by using a secondary objective function focused on idle times. Instances containing between 50 and 2000 requests were solved. The objective considered by Wong and Bell (2006) is the minimization of a linear combination of total operating time, passenger ride time and taxi cost for unassigned requests. The authors work with several vehicle types and maximum route durations. Some vehicles are equipped for wheelchair access but the capacities reserved for wheelchair users and non-wheelchair users are not substitutable. The authors propose a parallel insertion procedure. Users are first ranked according to an index that measures the difficulty and inconvenience caused to other requests when they are inserted into a route. The insertions are then made by considering the most difficult requests first. This is followed by a post-optimization phase consisting of trip reinsertions in other routes and trip exchanges. The algorithm was tested on artificial instances involving 150 requests. Wolfler Calvo and Colorni (2006) have devised a heuristic for a version of the DARP in which the number of available vehicles is fixed and windows are imposed on pickup and delivery 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.

Two exact branch-and-cut algorithms were recently proposed for the version of the DARP considered by Cordeau and Laporte (2003a). Cordeau (2006) and Ropke et al. (2007) both formulate the problem as an integer linear program (with some continuous variables in the first case). These two formulations are provided in Sects. 3.1 and 3.2, respectively. Several families of valid inequalities are also proposed for each model. Initially, the models are solved by relaxing some of the constraints. During the branching process, separation algorithms are applied to identify violated constraints among those that were initially relaxed or among the valid inequalities. These constraints are then introduced and the process ends when the search tree has been explored according to the usual branch-and-bound rules. The largest instance solved with the first algorithm contained 36 requests, while the largest one solved by the second algorithm contained 96 requests.

More recently, Melachrinoudis et al. (2007) have proposed a heuristic for a version of the DARP with soft time windows arising in a health-care organization. In this problem, the objective minimizes a linear combination of transportation costs and user inconvenience. Their heuristic uses tabu search with request reinsertions. Instances containing up to 50 requests were solved. Finally, Jørgensen et al. (2007) have worked with a multi-criteria objective containing seven terms. Their heuristic alternates between a genetic search mechanism to construct clusters and a modified nearest neighbour procedure to construct the routes. They have solved instances containing between 24 and 144 requests.

Table 2 summarizes the algorithms proposed for the multi-vehicle static DARP.

Table 2 Summary of several algorithms for the static multi-vehicle DARP

Reference	Objective	Time windows	Other constraints	Algorithm	Size of instances solved
Jaw et al. (1986)	Minimize non-linear combination of several types of disutility	On pickup or delivery	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)	Minimize weighted sum of differences between actual and desired delivery times, and differences between actual and shortest possible ride times	Upper bounds on pickup and delivery times	Vehicle capacity	Heuristic. Iterates between routing and scheduling phases	$n \approx 85$
Dumas et al. (1989a), Desrosiers et al. (1991), Ioachim et al. (1995)	Minimize number of vehicles used, then minimize total route duration	On pickup and delivery	Several vehicle types. Vehicle capacity. Maximum route duration	Heuristic. Create mini-clusters. Group them by column generation. Apply scheduling phase	$n \le 1890$ in DDS (1989), $n = 2411$ in DDSTV (1991), $n = 2545$ in IDDS (1995)
Toth and Vigo (1996, 1997)	Minimize total service cost	On pickup and delivery	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)	$276 \le n \le 312$
Borndörfer et al. (1997)	Minimize operational costs (drivers and vehicles)	On pickup and delivery	Several vehicle types. Vehicle capacity. Maximum route duration	Heuristic. Set partitioning formulation solved by truncated branch-and-cut algorithm	$859 \le n \le 1771$
Cordeau and Laporte (2003a)	Minimize total route length	On pickup or delivery	Vehicle capacity. Maximum route duration. Maximum ride time	Heuristic. Tabu search with vertex reinsertions	$24 \le n \le 295$

Table 2 (Continued)

Reference	Objective	Time windows	Other constraints	Algorithm	Size of instances solved
Aldaihani and Dessouky (2003)	Minimize total distance traveled by vehicles. Minimize total distance traveled by users	On pickup and delivery	Vehicle capacity	Initial insertion and reinsertion based heuristic. Tabu search	$42 \le n \le 155$
Diana and Dessouky (2004)	Minimize a weighted combination of distance, excess ride time over direct time, vehicle idle time	Lower bound on pickup time or upper bound on delivery time	Vehicle capacity. Maximum ride time. Maximum waiting time at pickup or drop-off location	Heuristic. Regret based insertion procedure	$n = 500$ and 1000
Rekiek et al. (2006)	Minimize the number of vehicles	On pickup and delivery	Vehicle capacity	Heuristic genetic algorithm for clustering. Insertions for routing	$100 \le n \le 164$
Xiang et al. (2006)	Minimize a linear combination of vehicle fixed and variable costs and of driver variable costs	On pickup and delivery	Several vehicle types. Vehicle capacity. Maximum route durations. Driver breaks and maximum working times	Heuristics. Insertions, inter-route exchanges. Secondary objective to provide diversification	$50 \le n \le 2000$
Wong and Bell (2006)	Minimize a linear combination of total operating time, passenger ride time and taxi cost for unassigned requests	On pickup and delivery	Several vehicle types. Vehicle capacity (some vehicles can accommodate wheelchairs). Maximum route duration. Maximum ride time proportional to direct travel time	Parallel insertions. Reinsertions and exchanges	$n = 150$

Table 2 (Continued)

Reference	Objective	Time windows	Other constraints	Algorithm	Size of instances solved
Wolfler Calvo and Colorni (2006)	Minimize a linear combination of number of serviced users and level of service of these users	On pickup and delivery	Vehicle capacity	Heuristic. Solve assignment problems, insert subtours in routes, make routes feasible	$10 \leq n \leq 180$
Cordeau (2006)	Minimize total route length	On pickup or delivery	Vehicle capacity. Maximum route duration. Maximum ride time	Exact branch-and-cut algorithm	$16 \leq n \leq 36$
Ropke et al. (2007)	Minimize total route length	On pickup or delivery	Vehicle capacity. Maximum route duration. Maximum ride time	Exact branch-and-cut algorithm	$16 \leq n \leq 96$
Melachrinoudis et al. (2007)	Minimize a linear combination of transportation cost and user inconvenience	On pickup and delivery. Soft and hard limits	Several vehicle types. Vehicle capacity	Heuristic. Tabu search with vertex reinsertions	$2 \leq n \leq 50$
Jørgensen et al. (2007)	Minimize a linear combination of transportation time, ride time, excess of maximum ride time, waiting time, time windows violations, work time and excess work time	On pickup and delivery	Vehicle capacity	Heuristic. Alternation between cluster construction by genetic search, and sequential route construction through nearest neighbour procedure	$24 \leq n \leq 144$

Table 3 Summary of several algorithms for the dynamic multi-vehicle DARP

Reference	Objective	Time windows	Other constraints	Algorithm	Size of instances solved
Madsen et al. (1995)	Multi-criteria objective	On pickup or delivery	Several vehicle types. Vehicle capacity. Maximum route duration. Maximum deviation between actual and shortest possible ride times	Heuristic. Vertex insertions	$n = 300$
Teodorovic and Radivojevic (2000)	Minimize a function incorporating route lengths, ride times and time window violations	On pickup and delivery	Vehicle capacity	Sequential insertion of users in vehicle routes. Nine rules are used to give more or less weight to the various elements of the objective	$n = 900$
Colorni and Righini (2001)	Maximize the number of serviced requests or maximize the perceived level of service, or minimize the total traveled distance	On pickup and delivery	Vehicle capacity. Maximum route duration	Alternation between clustering and routing algorithms. Branch-and-bound algorithm is applied to sequence a subset of users with time windows not too far in the future	None
Coslovich et al. (2006)	Minimize user dissatisfaction	On pickup and delivery	Deviation from desired service time. Upper bound on “excess ride time”	Insertions in current routes. Route reoptimizations with modified 2-opt	$25 \leq n \leq 50$

5.2 The dynamic case

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.

One interesting multi-vehicle dynamic DARP 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 pickup or delivery 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 instance. The authors report that the algorithm was capable of generating good quality solutions within very short computing times.

The problem solved by Teodorovic and Radivojevic (2000) does not work with a well-defined objective function, but with a set of three criteria to be minimized: the total traveled distance, vehicle waiting times, and passenger ride times. Whenever a new request arrives, it is inserted in one of the vehicle routes according to one of nine rules which depend on whether the insertion of the request in a given route would yield a "small," "medium" or "big" increase of traveled distance or vehicle waiting time. Depending on the outcome, the algorithm assigns a "very weak," "weak," "medium," "strong" or "very strong" preference to each vehicle. All numbers used in the evaluation are fuzzy and all preference measures are calculated by means of fuzzy functions. Similar rules are used to determine the position of the request in the selected route. The system was tested on artificial instances containing 900 requests.

Colorni and Righini (2001) have tested three different objectives, namely the maximization of serviced requests, the maximization of the perceived level of service by users, and the minimization of traveled distance. Their system assumes that a negotiation with users takes place in order to discourage them from imposing unduly tight time windows. The insertion mechanism alternates between a clustering phase and a routing phase. The routing algorithm applies branch-and-bound to a set of requests whose time windows are not too distant in the future. The clustering phase works with exchanges. The authors report that they have performed experiments with their system (DARIA) in Crema and Verbania, located in northern Italy, but they do not report any computational results.

A recent algorithm by Coslovich et al. (2006) follows a two-phase strategy for the insertion of a new request into an existing route. An off-line phase is first used to create a feasible neighborhood of the current route through a 2-opt mechanism. An on-line phase is then used to insert the new request with the objective of minimizing user dissatisfaction.

A summary of algorithms for the dynamic multi-vehicle DARP is presented in Table 3.

6 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 pickup 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. 2003).

Acknowledgement 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 227837-04 and 39682-05. This support is gratefully acknowledged. Thanks are also due to a referee who provided valuable comments.

References

Aldaihani, M., & Dessouky, M. M. (2003). Hybrid scheduling methods for paratransit operations. *Computers & Industrial Engineering*, *45*, 75–96.

Bodin, L. D., & 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. (2003). Ambulance location and relocation models. *European Journal of Operational Research*, *147*, 451–468.

Colorni, A., & Righini, G. (2001). Modeling and optimizing dynamic dial-a-ride problems. *International Transactions in Operational Research*, *8*, 155–166.

Cordeau, J.-F. (2006). A Branch-and-cut algorithm for the dial-a-ride problem. *Operations Research*, *54*, 573–586.

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

Cordeau, J.-F., & Laporte, G. (2003b). The dial-a-ride problem (DARP): variants, modeling issues and algorithms. *4OR: A Quarterly Journal of Operations Research*, *1*, 89–101.

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., Potvin, J.-Y., & Savelsbergh, M. W. P. (2007a). Transportation on demand. In C. Barnhart & G. Laporte (Eds.), *Transportation*. Amsterdam: Elsevier.

Cordeau, J.-F., Laporte, G., Savelsbergh, M. W. P., & Vigo, D. (2007b). Vehicle routing. In C. Barnhart & G. Laporte (Eds.), *Transportation*. Amsterdam: Elsevier.

Coslovich, L., Pesenti, R., & Ukovich, W. (2006). A two-phase insertion technique of unexpected customers for a dynamic dial-a-ride problem. *European Journal of Operational Research*, *175*, 1605–1615.

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, HEC Montréal.

Diana, M., & Dessouky, M. M. (2004). A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows. *Transportation Research Part B*, *38*, 539–557.

Dumas, Y., Desrosiers, J., & Soumis, F. (1989a). Large scale multi-vehicle dial-a-ride problems. Les Cahiers du GERAD, G-89-30, HEC 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, HEC 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, M. W. P. (2002). Efficient feasibility testing for dial-a-ride problems. *Operations Research Letters*, *30*, 169–173.

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

Jaw, 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 B*, *20*, 243–257.

Jørgensen, R. M., Larsen, J., & Bergvinsdottir, K. B. (2007, in press). Solving the dial-a-ride problem using genetic algorithms. *Journal of the Operational Research Society*.

Madsen, O. B. G., Ravn, H. F., & Rygaard, J. M. (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.

Melachrinoudis, E., Ilhan, A. B., & Min, H. (2007). A dial-a-ride problem for client transportation in a health-care organization. *Computers & Operations Research*, *34*, 742–759.

Mitrović-Minić, S., Krishnamurti, R., & Laporte, G. (2004). Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows. *Transportation Research B*, *38*, 669–685.

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

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

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 B. L. Golden & A. A. Assad (Eds.), *Vehicle routing: method and studies* (pp. 223–248). Amsterdam: North-Holland.

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

Rekiek, B., Delchambre, A., & Saleh, H. A. (2006). Handicapped person transportation: an application of the grouping genetic algorithm. *Engineering Application of Artificial Intelligence*, *19*, 511–520.

Ropke, S., Cordeau, J.-F., & Laporte, G. (2007). Models and branch-and-cut algorithms for pickup and delivery problems with time windows. *Networks*, *49*, 258–272.

Savelsbergh, M. W. P. (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, L. D. (1985a). Optimizing single vehicle many-to-many operations with desired delivery times: I. Scheduling. *Transportation Science*, *19*, 378–410.

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

Teodorovic, D., & Radivojevic, G. (2000). A fuzzy logic approach to dynamic dial-a-ride problem. *Fuzzy Sets and Systems*, *116*, 23–33.

Toth, P., & Vigo, D. (1996). Fast local search algorithms for the handicapped persons transportation problem. In I. H. Osman & J. P. Kelly (Eds.), *Meta-heuristics: theory and applications* (pp. 677–690). Boston: Kluwer.

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

Wolfler Calvo, R., & Colorni, A. (2007). An effective and fast heuristic for the dial-a-ride problem. *4OR: A Quarterly Journal of Operations Research*, *5*, 61–73.

Wong, K. I., & Bell, M. G. H. (2006). Solution of the dial-a-ride problem with multi-dimensional capacity constraints. *International Transactions in Operational Research*, *13*, 195–208.

Xiang, Z., Chu, C., & Chen, H. (2006). A fast heuristic for solving a large-scale static dial-a-ride problem under complex constraints. *European Journal of Operational Research*, *174*, 1117–1139.