Combining Multicriteria Analysis and Tabu Search for Dial-a-Ride Problems

Julie Paquette[1*], Jean-François Cordeau[1]
Gilbert Laporte[1] and Marta M. B. Pascoal[2,3]

[1] CIRRELT, HEC Montréal, 3000 chemin de la Côte-Sainte-Catherine, Montréal, H3T 2A7 Canada

[2] Departamento de Matemática da FCTUC,Apartado 3008, EC Santa Cruz, 3001-501 Coimbra, Portugal

[3] Institute for Systems Engineering and Computers – Coimbra (INESCC)

julie.2.paquette@hec.ca; jean-francois.cordeau@hec.ca;gilbert.laporte@cirrelt.ca; marta@mat.uc.pt

[*] Corresponding author: Julie Paquette, phone: + 1 514-340-6745, fax: + 1 514-340-6834

December 17, 2012

Abstract

In the *Dial-a-Ride Problem* (DARP) the aim is to design vehicle routes for a set of users who must be transported between given origin and destination pairs, subject to a variety of side constraints. The standard DARP objective is cost minimization. In addition to cost, the objectives considered in this paper include three terms related to quality of service. This gives rise to a multicriteria problem. The problem is solved by means of a flexible and simple metaheuristic which efficiently integrates the reference point method for multicriteria optimization within a tabu search mechanism. Extensive tests were performed on randomly generated data and on real-life data provided by a major transporter in the Montreal area. Results indicate that the algorithm can yield a rich set of non-dominated solutions. It can also be employed to determine good trade-offs between cost and quality of service.

Key-words: dial-a-ride problem, multicriteria analysis, quality of service, tabu search, reference point method.

1 Introduction

In the *Dial-A-Ride Problem* (DARP), several users formulate pickup and delivery requests for transportation between origins and destinations. The same user typically makes two requests during the same day: an outbound request from home to a destination and an inbound request for the return trip. These requests are satisfied by a fleet of vehicles based at a common depot. In the classical variant of the problem, the aim is to plan a set of minimum cost vehicle routes satisfying all requests (or as many of them as possible), under side constraints. These include vehicle capacity constraints, route duration constraints, time windows, and maximum ride time constraints. Time windows are usually imposed on the arrival time at destination for outbound trips, and on the departure time from origin for inbound trips. The problem is said to be static if all requests are known at the time of planning, and dynamic if these are gradually revealed over time (see Psaraftis 1995). This paper addresses the static case. For surveys on the DARP, we refer to Cordeau and Laporte (2007) and to Berbeglia et al. (2010).

The DARP arises in the management of on-demand transportation systems provided to elderly and disabled people in many large cities. Applications have been reported in Copenhagen (Madsen et al. 1995), Bologna (Toth and Vigo 1996, 1997), Berlin (Borndörfer et al. 1997), Crema and Verbania (Colorni and Righini 2001), Los Angeles County (Diana and Dessouky 2004), Brussels (Rekiek et al. 2006), Milan (Wolfler Calvo and Colorni 2007), and a mid-size US city (Karabuk 2009). It is expected that dial-a-ride services will gain in importance in the coming years due to the aging of the population and the trend toward the development of ambulatory health care services.

In addition to the classical cost minimization objective, several authors have incorporated quality of service considerations in the solution of the DARP. Beyond the satisfaction of time windows and maximum ride time constraints which are quite widespread, the most common quality of service criteria are the difference between actual and desired arrival time (Beaudry et al. 2010; Jørgensen et al. 2007; Melachrinoudis et al. 2007; Coslovich et al. 2006), waiting time during the ride (Jørgensen et al. 2007), waiting time before departure (Psaraftis 1980), total waiting time (Diana and Dessouky 2004), mean ride time (Parragh et al. 2009), excess of maximum ride time (Jørgensen et al. 2007), ratio of actual ride time to direct ride time (Wolfler Calvo and Colorni 2007), excess ride time over direct time (Jørgensen et al. 2007;

Melachrinoudis et al. 2007; Coslovich et al. 2006), time elapsed between the call and the arrival time (Wilson et al. 1976), and maximum number of stops while a user is on board (Armstrong and Garfinkel 1982).

Service criteria are usually handled as constraints or as terms in the objective function, which results in the generation of a single solution. Ideally, the problem should be solved within a multi-objective setting because it involves non-commensurate objectives. The multicriteria algorithm would thus produce a set of non-dominated solutions, i.e. a Pareto front. Each solution is represented by a vector whose elements are the values taken by the terms of the objective function. A solution weakly dominates another solution if it is better in at least one objective and not worse in any other objective. The image of a Pareto front in the objective space is also referred to as the non-dominated frontier. All solutions lying on the same frontier are incomparable with each other. A set of solutions is Pareto optimal if there does not exist any other set of solutions that weakly dominates it. To our knowledge, only two groups of authors have devised truly multicriteria algorithms capable of producing a set of non-dominated solutions. However, these two methods can only handle two criteria. Thus, Baugh et al. (1998) have developed a bicriteria simulated annealing heuristic in which the two criteria are travel costs and time window violations. Parragh et al. (2009) have used a variable neighborhood search heuristic coupled with path relinking to jointly minimize transportation cost and average ride time.

Our work follows the second approach, but can handle any number of objectives. In a first step, we have identified three quality of service criteria through an extensive survey conducted with the users of the Réseau de Transport de Longueuil (RTL) which operates a dial-a-ride service on the South Shore of Montreal (see Paquette et al. 2012). An initial contact was made with 857 users, among which 572 accepted to receive a postal questionnaire. In total, 331 filled questionnaires were returned, yielding a response rate of 38.6%. Common factor analyses and ANOVAs were performed on the collected data, and three measurable quality determinants emerged: the waiting time during the time window at the origin, the waiting time during the time window at the destination, and the ratio of the actual ride time over direct ride time. Each of these criteria can be incorporated within an optimization scheme for the daily routing and scheduling of vehicles, as we will show.

The contribution of this paper is the development of a flexible and efficient multicriteria algorithm incorporating a tabu search process, capable of producing within a single execution

a large set of non-dominated solutions with respect to a standard travel cost criterion and several quality of service criteria. Our algorithm is flexible in the sense that it can accommodate a wide variety of criteria and constraints. In particular, we consider two user types (ambulatory and wheelchair-bound), which results in two types of capacity, a heterogeneous fleet (minibuses, regular taxis, adapted taxis), and constraints related to drivers' breaks. Our algorithm is also of general applicability since it can be used to solve other versions of the DARP and other routing problems different from the DARP.

The main purpose of a multicriteria algorithm, such as the one developed in this paper, is to provide managers with a tool to better understand the trade-offs between costs and quality of service. This type of algorithm is normally used at a tactical level, to help the managers set the priorities of the provider, as defined by weights in an objective function. The algorithm can also be used at an operational level, using the weights chosen at the tactical level, to optimize the routing and scheduling of vehicles on a daily basis.

The remainder of this paper is organized as follows. Section 2 is devoted to the presentation of a mathematical formulation of the problem. The algorithm is described in Section 3, and computational results on artificial and real-life instances are presented in Section 4. Conclusions follow in Section 5.

2 Mathematical model

Our problem can be formulated as an integer linear program. We first introduce the standard model for the single criterion DARP, with n users of a single type and a heterogeneous fleet of m vehicles. Let $G = (V, A)$ be a directed graph where $V = \{v_0, v_1, ..., v_{2n+1}\}$ is the node set and $A = \{(v_i, v_j), v_i, v_j \in V, i \neq j\}$ is the arc set. Each request i consists of an origin node v_i and a destination node v_{n+i}. Node v_0 represents the depot from which the vehicles start their route, and node v_{2n+1} is the depot to which they must return. We denote by $V' = V \setminus \{v_0, v_{2n+1}\}$ the subset of nodes excluding the nodes associated with the depots, by V'_{OD} the subset of destination nodes associated to outbound trips, and by V'_{IO} the subset of origin nodes associated to inbound trips.

Vehicle k has a capacity equal to C^k and performs a route whose duration must not exceed T_k. Node $v_i \in V$ has a load equal to q_i, with $q_i = -q_{n+i}$, a service time d_i, and a time

window $[e_i, l_i]$. The time window of the depot node is $[0, T]$, where T is the length of the planning horizon. With each arc (v_i, v_j) are associated a travel cost c_{ij} and a travel time t_{ij}.

The route R_k of vehicle k is defined as the set of arcs it follows. In addition, $J_k = \{v_i | \exists (v_i, v_j) \in R_k, v_j \in V\}$ is the set of nodess visited by R_k. We define binary variables x_{ij}^k equal to 1 if and only if vehicle k travels on arc (v_i, v_j). Additional variables are required to define vehicle schedules. The arrival time of vehicle k at node v_i is denoted by A_i^k, the service time of vehicle k at node v_i is equal to $B_i^k \geq \max\{e_i, A_i^k\}$, and the departure time of vehicle k from v_i is equal to $D_i^k = B_i^k + d_i$. The waiting time at node i is positive only when the lower bound of the time window is greater than the arrival time: $W_i^k = \max\{0, B_i^k - A_i^k\}$. The ride time of user i on vehicle k is denoted by $H_i^k = B_{n+i}^k - D_i^k$. This variable may not exceed a maximal allowed ride time L. Finally, variable Q_i^k denotes the load of vehicle k immediately after visiting node v_i.

As in the standard DARP model introduced by Cordeau (2006), the objective is the minimization of the transportation costs. The standard model is as follows:

$$\text{Minimize} \sum_{v_i \in V} \sum_{v_j \in V} \sum_{k=1}^{m} c_{ij} x_{ij}^k \tag{1}$$

subject to:

$$\sum_{v_j = v_1}^{v_n} x_{0j}^k = 1 \qquad (k = 1, ..., m) \tag{2}$$

$$\sum_{v_j \in V'} x_{ji}^k - \sum_{v_j \in V'} x_{ij}^k = 0 \qquad (v_i \in V', k = 1, ..., m) \tag{3}$$

$$\sum_{v_i = v_{n+1}}^{v_{2n}} x_{i,2n+1}^k = 1 \qquad (k = 1, ..., m) \tag{4}$$

$$\sum_{v_j \in V'} \sum_{k=1}^{m} x_{ij}^k = 1 \qquad (v_i \in V') \tag{5}$$

$$\sum_{v_j \in V'} x_{ij}^k - \sum_{v_j \in V'} x_{n+i,j}^k = 0 \qquad (v_i \in \{v_1, ..., v_n\}, k = 1, ..., m) \tag{6}$$

$$B_j^k \geq (B_i^k + d_i + t_{ij}) x_{ij}^k \qquad (v_i, v_j \in V', k = 1, ..., m) \tag{7}$$

$$Q_j^k \geq (Q_i^k + q_j) x_{ij}^k \qquad (v_i, v_j \in V', k = 1, ..., m) \tag{8}$$

$$H_i^k = B_{n+i}^k - (B_i^k + d_i) \qquad (v_i \in \{v_1, ..., v_n\}, k = 1, ..., m) \tag{9}$$

$$\max\{0, q_i\} \leq Q_i^k \leq \min\{C^k, C^k + q_i\} \qquad (v_i \in V', k = 1, ..., m) \tag{10}$$

$$t_{i,n+i} \leq H_i^k \leq L \qquad (v_i \in \{v_1, ..., v_n\}, k = 1, ..., m) \tag{11}$$

$$B_{2n+1}^k - B_0^k \leq T_k \qquad (k = 1, ..., m) \tag{12}$$

$$e_i \leq B_i^k \leq l_i \qquad (v_i \in V', k = 1, ..., m) \tag{13}$$

$$B_0^k \geq 0 \qquad (k = 1, ..., m) \tag{14}$$

$$B_{2n+1}^k \leq T \qquad (k = 1, ..., m) \tag{15}$$

$$x_{ij}^k \in \{0, 1\} \qquad (v_i, v_j \in V, k = 1, ..., m). \tag{16}$$

In this model, constraints (2), (3) and (4) ensure that each route starts and ends at the depot. Constraints (5) and (6) guarantee that each node is visited exactly once and that the origin and destination of a request are visited by the same vehicle. The consistency of time and vehicle load variables is ensured by constraints (7) and (8), respectively. User ride times are defined by constraints (9). Constraints (10) ensure that vehicle capacity is always respected, and constraints (11) mean that user ride times never exceed L. The maximum length of a route is imposed by constraints (12), and time windows are enforced through constraints (13), (14) and (15).

We have modified the standard DARP model in three ways. First, we have used the same bicriteria model as proposed by Parragh et al. (2009) to enable a comparison of the effectiveness of our algorithm. The mean user ride time (MRT), as defined by equation (17) was thus used to measure user inconvenience as a second objective to pursue:

$$MRT = \sum_{i=1}^{n} H_i^k / n. \tag{17}$$

This model will be referred to as model 1.

The second and third models are modifications of the standard model in order to incorporate several features of real-life problems. These relate to the multiplicity of user types and vehicles, to the imposition of new constraints, and to the introduction of quality criteria in the objective function. When the user inconvenience terms are added up in one objective, the problem becomes a bicriteria optimization problem, yielding model 2. Finally, the model obtained when the user inconvenience terms are each considered as distinct objectives, will

be referred to as model 3. All modifications described hereafter are applicable to models 2 and 3.

Two types of users are usually served by adapted transportation services: ambulatory users and wheelchair-bound users. We thus consider different service times and we assume that each vehicle can accomodate up to C_1^k ambulatory users and C_2^k wheelchair users. No substitutions are possible between these two user types. The fleet used to serve these users is generally heterogeneous and the number of vehicles of each type (minibuses, regular taxis and adapted taxis) is equal to m_1, m_2 and m_3, respectively. The values of C_1^k and C_2^k can be different for each type of vehicle, which affects constraints (8) and (10). Finally, variables Q_{1i}^k and Q_{2i}^k denote the load of vehicle k for each type of users immediately after visiting node v_i. Constraints (8) and (10) are then duplicated to account for each user type:

$$Q_{1j}^k \geq (Q_{1i}^k + q_{1j})x_{ij}^k \qquad (v_i, v_j \in V', k = 1, ..., m) \tag{18}$$

$$Q_{2j}^k \geq (Q_{2i}^k + q_{2j})x_{ij}^k \qquad (v_i, v_j \in V', k = 1, ..., m) \tag{19}$$

$$\max\{0, q_{1i}\} \leq Q_{1i}^k \leq \min\{C_1^k, C_1^k + q_{1i}\} \qquad (v_i \in V', k = 1, ..., m) \tag{20}$$

$$\max\{0, q_{2i}\} \leq Q_{2i}^k \leq \min\{C_2^k, C_2^k + q_{2i}\} \qquad (v_i \in V', k = 1, ..., m). \tag{21}$$

Two ride times are defined according to the territory served by the provider. In our case study, a smaller maximum ride time L_1 is associated to trips within the territory, and a larger maximum ride time L_2 is imposed on trips going outside of it. This impacts constraints (9) and (11). As in Jaw et al. (1986), the vehicle is not allowed to be idle while carrying users, which means that it must either move, or the driver must be busy helping a user get on or off the vehicle. This constraint is enforced through the inclusion of the term

$$a(s) = \sum_{v_i \in V'} \sum_{v_j \in V'} \sum_{k=1}^{m} x_{ij}^k (Q_{1i}^k + Q_{2i}^k) W_j^k \tag{22}$$

in the objective function. The maximal duration of route k is equal to T_k, and drivers are entitled to one break of duration ν, positionned around the middle of their working day. This constraint can be modeled by creating fictitious users $v_{2n+2}, ..., v_{2n+2+m_1}$ with time windows $[e_i = b_0 + T_k/2 - 2\nu, l_i = b_0 + T_k/2]$. Each of these fictitious users is constrained to be served by a specific vehicle. Another constraint specifies that no user can be onboard the vehicle

while the driver takes his break:

$$b(s) = \sum_{v_i \in V'} \sum_{j=2n+2}^{2n+2+m_1} \sum_{k=1}^{m_1} x_{ij}^k (Q_{1i}^k + Q_{2i}^k) \leq 0. \tag{23}$$

In our algorithm, this constraint is relaxed and is treated as a penalty in the objective function. These constraints can be applied to one or all types of drivers.

The objective function of the provider contains four terms. The first represents the variable vehicle costs and is based on the tarification contract that the provider has with its subcontractors (minibus operators and taxis). This cost is calculated as

$$Cost = \sum_{k=1}^{m_1} c_1 (B_{2n+1}^k - B_0^k)/60 + \sum_{v_i \in V} \sum_{v_j \in V} \sum_{k=m_1+1}^{m} c_2 x_{ij}^k + \sum_{v_i \in V} \sum_{v_j \in V} \sum_{k=m_1+1}^{m} c_3 x_{ij}^k c_{ij}, \tag{24}$$

where c_1 is the hourly cost for a minibus, c_2 is the flat rate for a taxi ride, and c_3 is the cost per kilometer for a taxi ride.

The other three terms of the objective function are based on the results of our empirical study performed in cooperation with the users of the RTL service (Paquette et al. 2012). The information collected from the survey has enabled us to quantify and model the users' preferences and to represent their perceived level of service in different situations. The subsequent terms define *user inconvenience*.

The second term of the objective function represents the waiting time within the destination time window for an outbound trip. The upper limit of this time window is specified by the user but the lower limit is determined by the transporter. According to the results of our survey, most users prefer the vehicle to arrive at their destination around the middle of the time window. The inconvenience curve is therefore modeled through a quadratic function as follows:

$$Quality1 = \sum_{v_i \in V'_{OD}} \sum_{k=1}^{m} ((l_i - e_i)/2 - B_i^k)^2. \tag{25}$$

The third term of the objective function represents the waiting time within the origin time window for an inbound trip. The lower limit of the time window corresponds to the time requested by the user, but the upper limit is set by the transporter. In this case, our survey

indicates that most users prefer the vehicle to arrive close to the beginning of the time window. The more they wait, the less they perceive the quality of service as being satisfactory. We again use a quadratic inconvenience curve to measure this type of inconvenience:

$$Quality2 = \sum_{v_i \in V'_{IO}} \sum_{k=1}^{m} (B_i^k - e_i)^2. \tag{26}$$

Finally, the fourth term of the objective function relates to the users' ride time. Our survey results indicate that the inconvenience cost associated with ride time is quadratic with respect to the ratio of actual ride time over direct ride time:

$$Quality3 = \sum_{i=1}^{n} \sum_{k=1}^{m} (H_i^k / t_{i,n+i})^2. \tag{27}$$

These modifications were incorporated into the model to represent the constraints and objectives of the provider under study. Each of these four terms is considered as a distinct objective in model 3. However, in model 2, the three terms on user inconvenience are added up to yield a bicriteria problem. A weight of 50% is used for $Quality1$ and weights of 25% are used for $Quality2$ and $Quality3$ based on the users' preferences. It is clear, however, that other constraints or objectives could apply in different contexts.

3 Multicriteria tabu search algorithm

Numerous multicriteria optimization studies generalize classical single criterion problems. Considering several criteria is often the most natural way to model practical problems encountered in real-life applications. The approaches to these problems can be labeled as *a posteriori* decision making, when the decision maker articulates his preferences after a set of solutions is calculated; interactive decision making, when interactions between the decision maker and the optimization phase take place; and *a priori* decision making, when the preferences are aggregated before the solutions are computed. The latter type of approach seems to be one of the most popular, given that it allows solving a multicriteria problem as a single criterion problem after scalarization. Alternative methods for multicriteria problems are the ε-constraint method, which consists in reducing the number of objective functions by transforming them into constraints of the problem, goal programming, which aims at minimizing

the distance to an ideal point, and ranking methods, which list solutions in order of one of the objective functions, while their dominance is tested by comparing them to the solutions obtained earlier. Other research has focused on extending single criterion methods such as dynamic programming, branch-and-bound or branch-and-cut to the multicriteria context. Over the past 20 years, we have witnessed a rise in the development of metaheuristics for approximating the set of solutions, namely tabu search (Gandibleux et al. 1997) and simulated annealing (Alves and Clímaco 2000). Further details about these and other methods for multicriteria combinatorial optimization problems can be found in Ehrgott (2000), in Hansen (2000), in Jozefowiez et al. (2012) and in the survey of Ehrgott and Gandibleux (2002).

3.1 Heuristics for multicriteria routing problems

The specific case of multicriteria routing problems has been discussed and reviewed in Boffey (1996) and Jozefowiez et al. (2008). The techniques applied to this type of problem are generally similar to those mentioned earlier. Most research has concentrated on the bicriteria Traveling Salesman Problem (Gupta and Warburton 1986; Paquete and Stützle 2003; Angel et al. 2004, 2005; Lust and Jaszkiewicz 2010; Lust and Teghem 2010). As mentionned in the introduction, Baugh et al. (1998) and Parragh et al. (2009) have developed heuristics restricted to the bicriteria DARP.

3.2 Description of our multicriteria tabu search algorithm

A naïve implementation of a heuristic such as tabu search for a discrete multicriteria optimization problem would be to repeatedly and independently apply it to an instance by using each time a different set of criteria weights, and discarding dominated solutions at the end of the process. Such a scheme is likely to be impratical if each local search application is time-consuming and the number of weight combinations considered is large. This approach will be particularly inefficient in problems where generating an initial feasible solution is NP-complete. This is the case of the DARP which contains time windows. To circumvent this difficulty we have developed an efficient search mechanism consisting of a single thread, but within which the criteria weights are dynamically modified, and dominated solutions are

discarded along the way. Our algorithm combines some features of the tabu search heuristic of Cordeau and Laporte (2003) for the classical DARP, and of the multicriteria reference point method of Clímaco et al. (2006).

3.2.1 Single criterion tabu search algorithm for the DARP

The tabu search heuristic is initialized with a solution s_0 constructed by assigning each request to a randomly selected vehicle. This solution is not necessarily feasible. The objective function $f(s)$ minimizes the sum of the routing cost $c(s)$ and of the weighted penalty terms for each relaxed constraint: $q(s)$ for violations of capacity, $d(s)$ for violations of duration, $w(s)$ for violations of time windows, and $t(s)$ for violations of ride time constraints. Relaxing the constraints enables the algorithm to explore infeasible solutions during the search and thus reach better solutions than would otherwise be possible. The weights of the penalty terms, respectively α, β, γ, and τ, are self-adjusting positive parameters. Initially set to 1, these parameters are divided or multiplied by 1.5 at each iteration depending on whether the solution is feasible or infeasible with respect to the corresponding contraint. A new best solution s^* is identified whenever $f(s') < f(s^*)$, where s' is the new solution obtained after applying a move and s^* is the current best solution.

At iteration t, the current solution s_t can be partly described by an attribute set $U(s) = \{(i,k) : \text{request } i \text{ is assigned to vehicle } k\}$. To improve s_t, inter-route exchanges are performed at every iteration, and intra-route exchanges are performed every 10 iterations, or whenever a new best solution is identified. The neighboorhood $N(s)$ of a solution s is composed of all the solutions s' that can be obtained from s by removing the attribute (i,k) and replacing it with attribute (i,k'). Every solution in the neighboorhood of the current solution is evaluated and the best move is applied. To minimize route duration, the forward time slack (Savelsbergh 1992) is used to compute the impact of a move. This is the maximum time by which departure from the depot can be delayed without violating any time window on a vehicle route. The best move that minimizes the total increase in $f(s)$ is performed, and attribute (i,k) is then added to the tabu list for θ iterations. Through an aspiration criterion, the tabu status of an attribute can, however, be revoked if that would allow the search process to reach a solution of smaller cost than that of the best known solution having that attribute. To diversify the search, any solution $\overline{s} \in N(s)$ such that $f(\overline{s}) \geq f(s)$ is

penalized by the term $p(\overline{s}) = \lambda c(\overline{s})\sqrt{nm}\rho_{ik}$, where ρ_{ik} corresponds to the number of times the attribute (i, k) was added to a solution during the search, $\sqrt{nm}$ is a scaling factor, and λ is a non-negative parameter controlling the aggressiveness of the diversification.

3.2.2 Adaptation to multiple objectives and added constraints

We now explain the main changes made to this heuristic to handle multiple objectives and the newly added constraints in models 2 and 3. The initialization phase was modified to take advantage of the characteristics of the real-life problem which motivated this study and to yield an initial solution closer to feasibility. First, users in wheelchairs are only included in minibus or adapted taxi routes since regular taxis cannot accomodate this type of users. Second, user requests are ranked in non-decreasing order of the beginning of their time window. Each request is added to a route in a rotating manner. This enables the algorithm to generate balanced routes and to spread out users who have to be served at the same time in different routes. Three terms representing penalties associated with relaxed constraints from real-life problems have been added to the objective function. The first two terms $\alpha_1 q_1(s)$ and $\alpha_2 q_2(s)$ represent the penalties associated with the ambulatory and wheelchair capacities constraints ((20) and (21)), respectively. The third term $\delta a(s)$ ensures that the waiting time of users is limited while the vehicle is idle. The fourth term $\chi b(s)$ represents the fact that a user cannot be onboard the vehicle while the driver takes his break (constraints (23)). In the algorithm, the parameters α_1, α_2, δ and χ will be modified dynamically, as will the parameters β, γ and τ. This way of proceeding leads the search process through a mix of feasible and infeasible solutions and allows the use of simple exchange operators which do not guarantee feasibility.

The reference point method of Clímaco et al. (2006) was designed for multicriteria path problems in which the aim is to determine a set of non-dominated feasible solutions which constitute an approximation of the Pareto front. Potential solutions identified during the search process are evaluated through their distance from an ideal point defined as a vector whose elements are the minimal value of each objective, typically approximated by a heuristic. This point is generally infeasible, for otherwise it would correspond to a unique non-dominated solution. Figure 1 illustrates these concepts. The distance between a potential solution and the ideal point is computed through one of several metrics (typically

Manhattan, Euclidean or Tchebychev). We have used the Manhattan metric for its ease of implementation and also because we could find no evidence that the other two metrics yield better solutions.

The objective function we have used in model 1 is defined as the linear combination

$$\begin{aligned} f(s) &= \omega_0|Cost - Cost^*| \\ &\quad +\omega_1|MRT - MRT^*| \\ &\quad +\alpha_1 q_1(s) + \alpha_2 q_2(s) + \beta d(s) + \gamma w(s) + \tau t(s) + \delta a(s) + \chi b(s), \end{aligned} \tag{28}$$

where $(Cost^*, MRT^*)$ is the ideal point, and ω_h is the weight of criterion h.

The objective function we have used in model 2 is defined as the linear combination

$$\begin{aligned} f(s) &= \omega_0|Cost - Cost^*| + \omega_1(0.5 \times |Quality1 - Quality1^*| + 0.25 \times |Quality2 - Quality2^*| \\ &\quad +0.25 \times |Quality3 - Quality3^*|) + \alpha_1 q_1(s) + \alpha_2 q_2(s) + \beta d(s) + \gamma w(s) + \tau t(s) \\ &\quad +\delta a(s) + \chi b(s), \end{aligned} \tag{29}$$

where $(Cost^*, 0.5 \times Quality1^* + 0.25 \times Quality2^* + 0.25 \times Quality3^*)$ is the ideal point, and ω_h is the weight of criterion h.

The objective function we have used in model 3 is defined as the linear combination

$$\begin{aligned} f(s) &= \omega_0|Cost - Cost^*| + \omega_1|Quality1 - Quality1^*| + \omega_2|Quality2 - Quality2^*| \\ &\quad +\omega_3|Quality3 - Quality3^*| + \alpha_1 q_1(s) + \alpha_2 q_2(s) + \beta d(s) + \gamma w(s) + \tau t(s) \\ &\quad +\delta a(s) + \chi b(s), \end{aligned} \tag{30}$$

where $(Cost^*, Quality1^*, Quality2^*, Quality3^*)$ is the ideal point, and ω_h is the weight of criterion h.

During the search, every feasible solution is compared to those belonging to the pool ND of non-dominated solutions, even if $f(s') > f(s^*)$. This enables the algorithm to identify more non-dominated solutions than if the usual rule $f(s') < f(s^*)$ were applied.

The weights ω_h play two distinct roles. First, they are necessary to measure all terms of the objective function on a comparable scale. As suggested by Clímaco et al. (2006), we use the

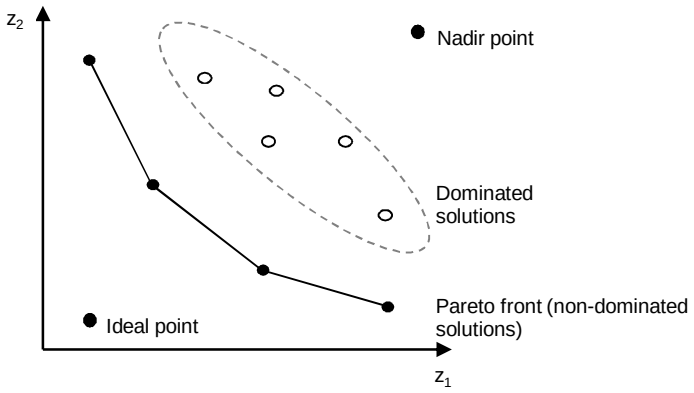

Figure 1: Representation of multicriteria concepts

scaling factors $\sigma_h = 1/(UB_h - LB_h)$, where UB_h and LB_h are the upper and lower bounds on the value of the term of the objective function corresponding to criterion h. These bounds are computed empirically by solving several instances repeatedly since no other information is available a priori. The weights ω_h also help perform a thorough exploration of the search space. To this end, they are updated whenever ι iterations have elapsed since the last non-dominated feasible solution was encountered, or whenever the algorithm has not identified any non-dominated feasible solution for more than ζ consecutive iterations. For model 3, the weights are updated as follows:

- $\omega_0 := \sigma_0\omega_0|Cost - Cost^*|$;
- $\omega_1 := \sigma_1\omega_1|Quality1 - Quality1^*|$;
- $\omega_2 := \sigma_2\omega_2|Quality2 - Quality2^*|$;
- $\omega_3 := \sigma_3\omega_3|Quality3 - Quality3^*|$;
- $\omega_h := \omega_h / \sum_{j=0}^{3} \omega_j \ (h = 0, ..., 3)$.

This procedure puts more emphasis on the term of the objective function for which the worst value was obtained compared to the ideal point on the common scale at the last iteration. At the same time, it allows the search process to put less emphasis on the other terms of the objective function, since the sum of weights is equal to one. It also allows the weights used in the previous iteration to be consistent from one update to the next. Whenever the weights are updated, the tabu list is emptied because it is no longer relevant with respect to the new weights.

Moreover, when optimizing model 3, the forward time slack procedure is not applied when $\omega_2 \geq 0.25$. Indeed, this procedure delays as much as possible the beginning of service at the nodes without violating any constraint, but since $Quality2$ aims at minimizing the gap between the beginning of the time window and the service time, it does not make sense to delay the start of service in this case.

Finally, the diversification procedure is usually based on the penalization of moves that are often used, by means of the penalty term $p(\overline{s})$. However, in the multicriteria algorithm, the modifications made to the weights ω_h also constitute a form of diversification. In the following section, tests will be executed to determine whether the standard diversification mechanism is still useful in this context. The resulting multicriteria algorithm is summarized in the pseudo-code of Algorithm 1.

Algorithm 1 Multicriteria algorithm

1. Initialization
2. Tabu search

Set $iterationbest = 1000000$, $iterationsolution = 1$, $iteration = 1$.

While $iteration < \eta$

1. If $iteration$ is a multiple of κ or if $newbest = 1$, do intra-route moves (intensification)
2. Otherwise
   (a) Evaluate the cost of all possible inter-route exchanges in the neighborhood of s_t.
   (b) Do the inter-route exchange that generates the best non tabu solution or that is permitted by the aspiration criteria and include the movement in the tabu list for θ iterations.
3. Calculate the costs for the solution found and the penalties associated with hard constraints.
4. Recalculate the parameters α_1, α_2, β, δ, γ, τ, χ.
5. If the solution is feasible
   (a) Compare s_t to solutions in the ND set.
   (b) If the solution is not dominated
      i. Add s_t to ND and reset tabu list.
      ii. Set $newbest = 1$, $iterationbest = 1$, and $iterationsolution = 1$.
   (c) If $iterationbest = \iota$ or if $iterationsolution = \zeta$
      i. Adjust the weights $\omega_0, \omega_1, \omega_2, \omega_3$.
      ii. Reset tabu list.
      iii. Set $iterationsolution = 1$.
6. Increment counters $iteration$, $iterationbest$ and $iterationsolution$.

4 Computational results

The algorithm just described was coded in C and run on a 2.93GHz Intel Xeon X7350 computer. It was tested on several random and real-life instances.

4.1 Test instances

Two sets of random instances were used for testing purposes. The first set was introduced by Cordeau (2006) and later used by Parragh et al. (2009) in a bi-objective context. This set contains two groups of 24 randomly generated instances. These instances contain between 16 and 96 requests and between two and eight vehicles. In group a, the vehicle capacity was set to $Q = 3$, only one passenger is picked-up at each origin ($q_i = 1$), and service time is three minutes for every user ($d_i = 3$). In group b, the instances have a vehicle capacity of $Q = 6$, every request consists of up to six passengers ($q_i \in 1, 2, 3, 4, 5, 6$) and service time is equal to the number of passengers ($d_i = q_i$). For both groups, the routing costs and travel times correspond to the Euclidean distance between the nodes which have coordinates in the $[-10, 10]^2$ square. Tight time windows are imposed on the destination node for outbound requests (l_{n+i} was chosen in the interval $[60, T]$, and $e_{n+i} = l_{n+i} - 15$). For inbound requests, tight time windows are imposed on the origin node (e_i was chosen in the interval $[0, T - 60]$, and $l_i = e_i + 15$). Each instance is identified with a letter corresponding to its group, the number of vehicles and the number of requests, i.e. a5-50 has 50 requests to serve with five vehicles, each having a capacity of three passengers. Model 1 applies to these instances.

The second set of random instances are derived from the first set of instances introduced by Cordeau and Laporte (2003), modified to include two types of users and two types of vehicles. We did not consider the second set of instances of these authors because their time windows are much wider than those of the real-life instances. The instances contain between 24 and 144 requests randomly generated over the square $[-10, 10]^2$. The maximum ride time L is equal to 90 minutes and the maximal duration of a route is set to 480 minutes. Since approximately 30% of the rides are performed by wheelchair users, this proportion was also used in our instance generation procedure. Service time for ambulatory users is eight minutes ($d_i = 8$) and 12 minutes for wheelchair users ($d_i = 12$). For each request, only one user boards ($q_i = 1$) or leaves the vehicle ($q_i = -1$). Thus, the total number n of

users is comprised of n_1 ambulatory users and n_2 wheelchair-bound users. A time window is associated to each origin node for an inbound trip and to each destination node for an outbound trip. The beginning e_i of the time window is randomly chosen in the interval [60, 480], and its end l_i is chosen in the interval $[e_i + 30, e_i + 45]$. Table 1 summarizes the characteristics of the random instances, where n represents the number of requests, $m1$ the number of minibuses and $m2$ the number of regular taxis available. Model 2 applies to these instances.

Table 1: Characteristics of the random instances

Instance	n_1	n_2	m_1	m_2	Instance	n_1	n_2	m_1	m_2
1	20	4	1	2	6	99	45	5	8
2	33	15	2	3	7	20	16	2	2
3	48	24	3	4	8	49	23	3	3
4	65	31	3	6	9	80	28	3	5
5	85	35	5	7	10	91	53	5	6

The real-life instances are those of the RTL. We had access to a full year of data. The daily number of rides changes according to the season and the day of the week. The peak demand occurs on Fridays, while weekends are the least busy days. To perform our tests, we have selected 12 typical days from two different months and representing all days of the week. Each of these instances has a different number of requests and a different number of vehicles available. Table 2 presents the main characteristics of the real-life instances. The service times for each type of user are estimated by the provider as $d_{1i} = 1$ minute and $d_{2i} = 2$ minutes. For each request, only one user boards ($q_i = 1$) or leaves the vehicle ($q_i = -1$). The values of C_1^k and C_2^k are 8 and 5 for the m_1 available minibuses, 4 and 0 for the m_2 available regular taxis, and 1 and 1 for the m_3 available adapted taxis. The RTL network extends over the South Shore of Montreal, where the depot is located, Montreal Island and the city of Laval, north of Montreal. If a trip is made on the South Shore, the maximum ride time is $L_1 = 45$ minutes; otherwise, it is equal to $L_2 = 90$ minutes. Time windows for outbound requests have a width of 20 minutes, whereas the width for inbound requests is 30 minutes. Minibus drivers cannot work more than 10.5 consecutive hours ($T_k = 630$) and are entitled to a break of $\nu = 30$ minutes, usually around the middle of their working day. The last two constraints do not apply to taxi drivers who serve other clients when they are not working

for the RTL. The hourly cost c_1 for a minibus is estimated at \$50, the fixed cost c_2 for a taxi ride is \$3.30, and the cost per kilometer c_3 is \$1.60. Model 3 applies to these instances.

Table 2: Characteristics of the real-life instances

Instance	Day of the week	n_1	n_2	m_1	m_2	m_3
February 5	Monday	816	240	22	50	8
February 6	Tuesday	874	262	22	50	8
February 7	Wednesday	875	271	22	50	9
February 8	Thursday	871	271	22	50	5
February 9	Friday	926	274	22	50	12
February 10	Saturday	351	103	19	50	5
February 11	Sunday	174	102	19	50	5
August 6	Monday	676	178	22	50	5
August 7	Tuesday	780	190	22	50	5
August 8	Wednesday	769	217	22	50	8
August 9	Thursday	758	222	22	50	5
August 10	Friday	688	194	19	50	5
August 11	Saturday	137	69	19	50	5
August 12	Sunday	121	69	19	30	5

4.2 Solution quality indicators

To measure solution quality, since we have to compare sets of solutions, we have used two indicators frequently employed in the literature on evolutionary algorithms (Zitzler et al. 2003, 2010). The first is the *hypervolume indicator* which measures the volume formed by the nadir point and all points in the set of non-dominated solutions, as shown in Figure 2 for sets 1 and 2. The hypervolume is thus the objective space that is weakly dominated by a set of solutions. In Figure 2 the hypervolume weakly dominated by solution set 1 corresponds to the dark grey region, the hypervolume weakly dominated by solution set 2 corresponds to the light grey region. We have used the algorithm developped by Fonseca et al. (2006) to compute the hypervolume. The larger the value of the indicator, the better is the set of solutions because this means it is diversified, which is a desirable property. For example,

in Figure 2, the solutions of set 2 are better than those of set 1. Our second indicator is the *multiplicative unary epsilon indicator*. It represents the smallest value of ϵ by which each point in a reference set has to be multiplied to obtain a set that is dominated by the set of non-dominated solutions. As suggested by Parragh et al. (2009), the reference set corresponds in this case to the combination of all non-dominated solutions identified for the same instance accross all runs executed on it. A small value of the indicator is preferable because this means that the set of solutions found by the algorithm is close to the reference set.

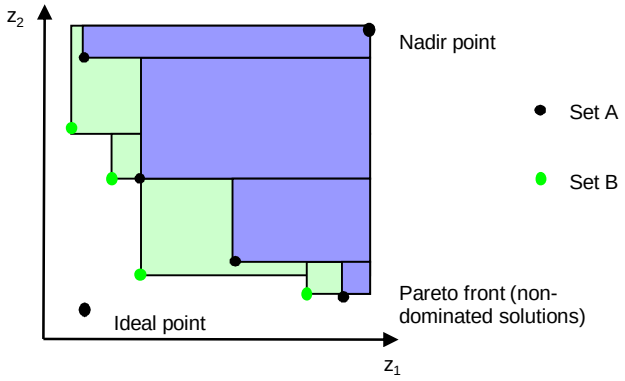

Figure 2: Hypervolume indicator for two sets of solutions

4.3 Preliminary tests on the random instances

Preliminary tests to parameterize the algorithm were executed with the random instances, aiming to

1. identify the best values for ι and ζ, the numbers of iterations after which the weights are changed,
2. determine whether the algorithm is efficient, and
3. identify the best value for δ, the coefficient of the penalty term representing the waiting time of users onboard the vehicle while it is idle.

Tests on 12 of the 48 instances from the first set of instances using model 1 were used to identify the best values of ι and ζ. Each instance was solved 10 times with 10 different seeds for 10^5 iterations. The algorithm was tested for the following 12 combinations of

values of (ι, ζ): (100, 500), (100, 1000), (100, 5000), (500, 1000), (500, 5000), (500, 10000), (1000, 5000), (1000, 10000), (1000, 50000), (5000, 10000), (5000, 50000) and (10000, 50000). The values of the lower and upper bounds for each optimization criterion were provided by Parragh et al. (2009) and are reported in Table 3.

Table 3: Values of the lower and upper bound for the twelve instances from the first set of instances

Instance	$\underline{Cost}$	$\underline{Quality}$	$\overline{Cost}$	$\overline{Quality}$	Instance	$\underline{Cost}$	$\underline{Quality}$	$\overline{Cost}$	$\overline{Quality}$
a2-16	294.94	503	11.71	30	b2-16	309.40	534.449	12.77	45
a2-20	344.83	640.93	10.28	30	b2-20	0	728.17	0	45
a2-24	431.12	756	11.85	30	b2-24	444.70	841.67	11.75	45
a3-18	300.48	563.62	10.63	30	b3-18	301.64	615.52	10.68	45
a3-24	344.83	685.29	9.73	30	b3-24	394.50	873.83	10.37	45
a3-30	494.84	946.82	12.90	30	b3-30	531.44	1132.22	12.47	45
a3-36	583.18	1184.34	11.60	30	b3-36	603.79	1308.6	10.86	45
a4-16	282.67	495.74	11.27	30	b4-16	296.95	585.36	12.08	45
a4-24	375.02	764.87	10.85	30	b4-24	371.41	825.46	9.09	45
a4-32	485.49	1003.9	10.88	30	b4-32	494.82	1251.72	10.15	45
a4-40	557.68	1322.87	10.40	30	b4-40	656.63	1461.55	10.42	45
a4-48	672.58	1449.62	10.11	30	b4-48	674.20	1747.06	9.52	45
a5-40	498.40	1237.16	9.72	30	b5-40	613.72	1567.49	11.32	45
a5-50	687.94	1578.76	10.70	30	b5-50	762.84	1941.99	11.08	45
a5-60	814.32	1994.12	11.02	30	b5-60	905.05	2435.9	10.88	45
a6-48	604.48	1549.38	10.13	30	b6-48	714.88	1828.96	10.96	45
a6-60	826.82	2009.48	11.32	30	b6-60	860.27	2305.85	10.99	45
a6-72	925.97	2373.78	10.36	30	b6-72	982.18	2721.49	10.40	45
a7-56	727.25	1866.73	10.64	30	b7-56	829.61	2087.04	10.92	45
a7-70	898.33	2282.58	10.67	30	b7-70	913.99	2690.21	9.29	45
a7-84	1044.15	2804.51	10.93	30	b7-84	1223.57	3303.62	11.47	45
a8-64	748.93	2073.85	9.96	30	b8-64	846.75	2360.51	9.64	45
a8-80	955.63	2578.25	9.75	30	b8-80	1051.2	2987.97	10.01	45
a8-96	1242.44	3282.66	11.10	30	b8-96	1205.83	3587.1	10.02	45

Table 4 presents the average of the hypervolume and epsilon indicators on 10 runs for all 12 instances for the different parameterizations based on ι and ζ values. According to the hypervolume indicator, the best parameterization is the one with $\iota = 1000$ and $\zeta = 5000$.

However, according to the epsilon indicator, the best parameterization would be $\iota = 100$ and $\zeta = 500$. If we normalize the results for each indicator and add the sum of both indicators normalized as shown in the last three columns of Table 4, the best parameterization is $(\iota = 1000, \zeta = 5000)$ since it has the best value for the hypervolume indicator and is not too far from the best value for the epsilon indicator.

Table 4: Indicator values for parameters ι and ζ (average on all instances and runs)

ι	ζ	Hypervolume indicator	Epsilon indicator	Normalized hypervolume indicator	Normalized epsilon indicator	Sum of normalized indicators
100	500	26616.97	**1.08342**	0.3991	0.0000	0.3991
100	1000	26615.35	1.08397	0.4017	0.0109	0.4126
100	5000	26551.69	1.08478	0.5042	0.0270	0.5312
500	1000	26815.72	1.08367	0.0792	0.0049	0.0841
500	5000	26774.70	1.08414	0.1452	0.0142	0.1592
500	10000	26757.79	1.08473	0.1724	0.0259	0.1724
1000	5000	**26864.91**	1.08451	0.0000	0.0216	**0.0216**
1000	10000	26854.88	1.08442	0.0161	0.0198	0.0359
1000	50000	26785.85	1.08777	0.1273	0.0861	0.2134
5000	10000	26821.52	1.09139	0.0698	0.1579	0.2277
5000	50000	26812.74	1.09222	0.0840	0.742	0.2582
10000	50000	26243.69	1.13390	1.0000	1.0000	2.0000

Tests on the other 36 instances from the first set were performed to determine whether the algorithm is efficient. Instances were each run 10 times for 10^5 iterations. Aggregated results for groups a and b are provided in Table 5 and compared to the results obtained by Parragh et al. (2009). The CPU time is in minutes. We observe that our algorithm is slightly less efficient than that of Parragh et al. However, it is simpler and more flexible in the sense that it can easily incorporate additional constraints and objectives, which is not straightforward to implement in the Parragh et al. algorithm.

Table 5: Indicator values for 36 instances (average on all instances and runs)

	Our algorithm			Parragh et al. phase 1 algorithm			Parragh et al. phase 2 algorithm		
Instance	Epsilon	Hypervolume	CPU	Epsilon	Hypervolume	CPU	Epsilon	Hypervolume	CPU
Avg. group a	1.07861	0.32969	21.66	1.05094	0.33988	4.83	1.01475	0.34181	5.08
Avg. group b	1.06768	0.43612	27.30	1.03028	0.44184	7.94	1.01168	0.44245	8.19
Global avg.	1.07330	0.38139	24.40	1.04091	0.38940	6.34	1.01326	0.39069	6.59

Tests on the second set of instances with model 2 were then executed to identify the best value for δ, which is a parameter found only in models 2 and 3 since it is linked to an added constraint. Each random instance was solved 10 times with 10 different seeds for 10^5 iterations. Three values were tested for δ: 0.5, 1, and 1.5. The values of the lower and

upper bounds for each optimization criterion are summarized in Table 6. These values are an approximation of the best values found when individually optimizing each objective term five times.

Table 6: Values of the lower and upper bound for the random instances

Instance	$\underline{Cost}$	$\underline{Quality1}$	$\underline{Quality2}$	$\underline{Quality3}$	$\overline{Cost}$	$\overline{Quality1}$	$\overline{Quality2}$	$\overline{Quality3}$
pr01	400	0	0	24	900	3900	7700	25800
pr02	900	0	0	48	1400	5000	15300	117500
pr03	1530	0	0	72	2200	5300	20500	12000
pr04	1780	0	0	96	2700	9500	28700	96000
pr05	2100	0	0	120	3200	10500	28300	42600
pr06	2710	0	0	144	3900	11500	37800	75600
pr07	780	0	0	36	1200	4000	11400	64900
pr08	1480	0	0	72	2200	7600	17800	12800
pr09	1980	0	0	108	2800	9200	25800	25700
pr10	2990	0	0	144	4000	10700	37900	7468700

Table 7 presents aggregated results over the values of δ. The best value is 0.5 according to both indicators.

Table 7: Indicator values associated to the penalty for waiting time of users onboard the vehicle while it is idle (average on all instances)

δ	Hypervolume indicator	Epsilon indicator
0.5	20241818.70	1.19067
1	19978775.99	1.19068
1.5	18906322.80	1.19478

4.4 Application to the real-life instances

Having parameterized the algorithm on the random instances, we have applied it with the best parameter settings to the real-life instances on the bi-objective problem (model 2) and on the multicriteria problem (model 3). To determine the upper and lower bounds necessary to set the scales of each objective term, the algorithm was run separately on weekday instances

and on weekend instances. In a context where this method of analysis should be repeated, it is important to consider that the determination of scales is an initial step that cannot be executed for each new instance, i.e. everyday. Thus, all weekday instances will have the same scales, and all weekend instances will have others (see Table 8 and 9). We have used the parameter values derived from our tests on the random instances, i.e. $\iota = 1000$, $\zeta = 5000$ and $\delta = 0.5$.

Table 8: Upper and lower bounds for real-life instances from Monday to Friday

$\underline{Cost}$	$\underline{Quality1}$	$\underline{Quality2}$	$\underline{Quality3}$	$\overline{Cost}$	$\overline{Quality1}$	$\overline{Quality2}$	$\overline{Quality3}$
8000	0	0	854	30000	60000	300000	20000

Table 9: Upper and lower bounds for real-life instances on Saturday and Sunday

$\underline{Cost}$	$\underline{Quality1}$	$\underline{Quality2}$	$\underline{Quality3}$	$\overline{Cost}$	$\overline{Quality1}$	$\overline{Quality2}$	$\overline{Quality3}$
3000	0	0	190	20000	20000	200000	10000

Table 10 presents for each instance the number of non-dominated solutions found, the values of the objective terms for the best solution according to the first objective, the values of the objective terms for the best solution according to the second objective, the correlation coefficient between both objectives, and the CPU time in minutes.

Table 10: Computational results for model 2 on the real-life instances

		Solution with best cost value		Solution with best user inconvenience value			
Instance	Number of non-dominated solutions	Cost	User inconvenience	Cost	User inconvenience	Correlation coefficient	CPU
February 5	56	11675.42	15636.68	12510.00	11346.04	−0.9415	6901.85
February 6	14	11815.18	11564.43	11988.6	10724.45	−0.8577	9145.43
February 7	24	11791.46	11885.34	12225.27	10880.30	−0.9614	10080.00
February 8	21	12160.66	13370.44	12430.86	12338.76	−0.9436	9864.36
February 9							
February 10	27	5390.73	2236.48	4678.38	2499.60	−0.7915	229.59
February 11	39	4007.47	2388.84	5088.81	1252.76	−0.9582	229.59
August 6	90	8941.17	9485.75	10766.16	5461.14	−0.9385	4774.92
August 7	19	10766.77	7752.71	11185.87	7252.04	−0.9312	5098.33
August 8	40	10427.80	11719.92	11902.30	8011.84	−0.9167	6984.63
August 9	58	10601.66	10821.79	11149.58	8410.73	−0.9401	8081.81
August 10	37	9358.87	10735.53	10854.21	5812.43	−0.8942	5791.81
August 11	20	3234.41	1623.85	4483.10	1062.12	−0.9684	115.57
August 12	36	3856.22	1510.02	3004.16	4858.28	−0.8611	192.11

The value of the correlation coefficient between normalized cost and normalized user inconvenience on all instances is −0.8795, which indicates a strong negative relationship between

both variables. If raw values of these variables were taken, a positive correlation would be obtained since the cost and the user inconvenience increase with the size of the instances. Therefore, to remove the size of instances from the analysis, normalized values were correlated.

To illustrate how managers could use the information provided by the algorithm, consider Figure 3 which represents the solution set for the August 6 instance. Based on these solutions, we can compute a regression to explain cost with user inconvenience. To ensure that each instance is equally represented in the analysis, we have chosen 14 solutions from the total solution set obtained for each instance. The two extreme solutions were kept and 12 solutions were chosen randomly for each instance. The regression was run on the data obtained from the 196 solutions. With this regression, we want to explain the impact of an increase of one unit of user inconvenience (the independent variable) on the cost (the dependent variable). To remove the impact of instance size, 13 dummy variables were created (if we create 14 variables, one of the variable will be a linear combination of the others, which will result in multicolinearity). The results of the regression are shown in Table 11. The regression has an R-square value of 0.9963, which is near perfection. Thus, we can conclude that an increase of one unit in user inconvenience will decrease cost by \$0.31. Another regression where user inconvenience is the dependent variable and user inconvenience is the dependent variable was run. The R-square value is again really good at 0.9825 and we can conclude that an increase of one unit in cost will decrease user inconvenience by 2.28 units. We can conclude from these results that it is not too costly to provide a better quality of service. However, since user inconvenience is an aggregation of three measures, it is difficult to really evaluate the meaning of variations on the quality of service. We will pursue our analysis using the results from model 3.

Figure 3: Solution set for the August 6 instance

Table 11: Regression results performed on 196 solutions obtained from all instances where cost is the dependent variable

Variable	Parameter estimate	Standard error	t value	Significance level
intercept	4419.45143	79.59972	55.52	< 0.0001
quality	-0.31243	0.01527	-20.46	< 0.0001
inst5feb	11854	174.79328	67.82	< 0.0001
inst6feb	10919	139.53350	78.25	< 0.0001
inst7feb	11139	141.63568	78.65	< 0.0001
inst8feb	11899	160.96983	73.92	< 0.0001
inst10feb	1261.90577	82.75376	15.25	< 0.0001
inst11feb	599.15971	86.05504	6.96	< 0.0001
inst6aug	7559.30266	96.89984	78.01	< 0.0001
inst7aug	8909.29068	99.96798	89.12	< 0.0001
inst8aug	9646.56346	120.53019	80.03	< 0.0001
inst9aug	9644.62634	125.88840	76.61	< 0.0001
inst10aug	8168.16780	104.41390	78.23	< 0.0001
inst11aug	-10.31114	88.11749	-0.12	0.9070

Tests were further run on the 14 real-life instances using model 3 and the parameter values derived from our tests on the random instances. Table 12 presents for each instance the number of non-dominated solutions found, the best and worse values obtained for each objective, and the CPU time in minutes. Comparing the results of models 3 (Table 12) with those of model 2 (Table 10), it appears that the latter model is more effective in identifying low cost solutions. This can be partly explained by the fact that model 2 leads to the identification of more feasible and new best solutions, which in turns yields more intensification (intra-route exchanges). Because intra-route exchanges are one of the most time consuming components of the algorithm, this results in larger computing times for model 2.

Table 12: Computational results for Model 3 on the real-life instances

Instance	Number of non-dominated solutions	*Cost* best value	*Cost* worst value	*Quality*1 best value	*Quality*1 worst value	*Quality*2 best value	*Quality*2 worst value	*Quality*3 best value	*Quality*3 worst value	CPU
February 5	290	12511.16	12818.44	8473.96	11211.15	26447.14	42698.22	2093.85	2801.25	4773.62
February 6	71	12931.23	13162.30	9960.19	13400.42	21817.32	57747.18	2876.87	3237.14	3489.14
February 7	28	12440.74	12548.31	10681.06	11302.49	29157.04	31291.55	2977.10	3081.89	3666.03
February 8	279	12518.40	13161.59	8883.28	14147.46	19608.92	62349.65	2692.29	3998.27	7535.89
February 9										
February 10	36	8042.84	9748.05	4600	15088.13	342.49	185039.55	843.76	2688.12	1335.15
February 11	681	4373.36	7946.48	1565.74	11898.85	305.77	70798.30	424.69	3058.63	222.83
August 6	426	10164.83	11893.17	7437.06	9104.91	6270.69	29380.11	1529.74	2444.62	3100.03
August 7	315	10165.57	12606.29	8713.14	11674.78	4213.92	17467.65	1791.62	2969.26	5103.51
August 8	157	11746.43	12335.28	10766.60	12462.25	6019.11	43743.82	2270.05	3780.85	3216.16
August 9	418	11330.27	12360.94	7413.48	11179.90	9127.15	35977.44	2008.02	2899.60	4453.15
August 10	353	9417.03	11192.44	7583.25	10285.86	8650.11	30264.20	1816.21	2734.68	3125.97
August 11	54	4722.36	5479.13	1649.27	2988.89	0	4.14	685.83	1228.68	165.97
August 12	678	3827.11	4736.76	1121.16	4190.72	220.39	3122.86	267.30	1286.2	96.94

We have made a three step analysis on these results. We have used 28 solutions per instance. All extreme solutions (on one of the four objectives) were retained and 20 solutions per instance were also chosen randomly. Again, this is to ensure that all instances will be equally represented.

First, correlations between cost *Quality*1, *Quality*2 and *Quality*3 were computed as presented in Table 13. Again, normalized values were used to remove the impact of instances size since we wish to understand the relationship between cost and user inconvenience measures, not how these variable behave for different demand sizes. We can conclude that *Cost* is positively correlated to *Quality*1 and *Quality*3 and that *Quality*2 is negatively correlated

to all variables. This result may seem surprising at first since one would normaly expect user inconvenience measures to be negatively correlated with $Cost$.

Table 13: Correlations between $Cost$, $Quality1$, $Quality2$ and $Quality3$

	$Cost$	$Quality1$	$Quality2$	$Quality3$
$Cost$	1	0.2359	−0.4867	0.5476
$Quality1$	0.2359	1	−0.2443	0.3313
$Quality2$	−0.4867	−0.2443	1	−0.7652
$Quality3$	0.5476	0.3313	−0.7652	1

Second, to further analyse the results, canonical correlations were run. Canonical correlations measure the strength of the relation between one variable and the best linear combination of the other variables. This allows to see the impact of many varibles on only one while keeping them separate at the same time. Table 14 summarizes the results of this analysis. All correlations are quite high except for the one with $Quality1$.

Table 14: Correlations between $Cost$, $Quality1$, $Quality2$ and $Quality3$

Dependent variable	Independent variable	Canonical correlation coefficient
$Cost$	$Quality1$, $Quality2$, $Quality3$	0.5608
$Quality1$	$Cost$, $Quality2$, $Quality3$	0.3384
$Quality2$	$Cost$, $Quality1$, $Quality3$	0.7696
$Quality3$	$Cost$, $Quality1$, $Quality2$	0.8004

Third, regression analyses were performed to better understand the trade-offs between cost and user inconvenience measures, and between the different user inconvenience measures. Four groups of regression were run on the 392 solutions, each using one of the four variable as the independent variable explaning the three other variables. For all the regressions, 13 dummy variables were also used as independent variables to remove the impact of size and particularities of the different instances. To simplify the presentation of the results, the coefficient for the dummy variables are not presented since they are not relevant to our analysis. The results of this analysis are summarized in Table 15. We have favored this approach over a single multiple regression analysis since separate single regressions allow for a better interpretation of the coefficients in the presence of multicollinearity.

Table 15: Regressions between $Cost$, $Quality1$, $Quality2$ and $Quality3$

Dependent variable	Coefficient of the independent variable	Independent variable	R-square
$Quality1$	0.3240	$Cost$	0.7593
$Quality2$	-15.4551	$Cost$	0.3917
$Quality3$	0.2641	$Cost$	0.8311
$Cost$	0.0298	$Quality1$	0.9711
$Quality2$	10.7452	$Quality1$	0.6920
$Quality3$	-0.0591	$Quality1$	0.8207
$Cost$	-0.0067	$Quality2$	0.9739
$Quality1$	0.0509	$Quality2$	0.8898
$Quality3$	-0.1172	$Quality2$	0.9120
$Cost$	0.4528	$Quality3$	0.9743
$Quality1$	-1.1023	$Quality3$	0.7728
$Quality2$	-46.1785	$Quality3$	0.6885

5 Conclusions

We have developed a multicriteria heuristic embedding a tabu search process in order to solve DARPs combining cost and quality of service criteria. To our knowledge, our algorithm is the first that can handle more than two criteria for this type of problem. It provides a useful tool to assist decision making in a practical context. Our results indicate that computing times are acceptable, considering that this type of planning is often made at the tactical level. The algorithm generates many high quality non-dominated solutions to real-life DARPs. We have performed extensive tests to analyze the interactions among several criteria including solution cost and quality of service indicators. Using the set of solutions generated by the algorithm, managers can readily evaluate the impacts of potential policy changes on the service performance. Finally, the algorithm is flexible and other constraints or criteria that could arise in other contexts can readily be incorporated within the proposed solution scheme.

Acknowledgements

This work was partly supported by the Canadian Social Sciences and Humanities Research Council, by the Canadian Natural Sciences and Engineering Research Council under grants 227837-09 and 39682-10, and by the Portuguese FCT grant SFRH/BSAB/830/2008 and project POSC/U308/1.3/NRE/04. This support is gratefully acknowledged. We also thank Marc Fredette for his help with the analysis of the results of model 3, and the anonymous referees for their valuable comments.

References

Alves, M. J. and J. Clímaco (2000). An interactive method for 0-1 multiobjective problems using simulated annealing and tabu search. *Journal of Heuristics* 6, 385–403.

Angel, E., E. Bampis, and L. Gourvés (2004). Approximating the Pareto curve with local search for the bicriteria TSP(1,2) problem. *Theoretical Computer Science* 310, 135–136.

Angel, E., E. Bampis, L. Gourvés, and J. Monnot (2005). (Non)-approximability for the multi-criteria TSP(1,2). In M. Liskiewicz and R. Reischuk (Eds.), *Fundamentals of Computation Theory*, Volume 3623 of *Lecture Notes in Computer Science*, pp. 329–340. Berlin: Springer.

Armstrong, G. R. and R. S. Garfinkel (1982). Dynamic programming solution of the single and multiple vehicle pickup and delivery problem with application to dial-a-ride. Working paper 162, University of Tennessee.

Baugh, J. W. J., G. K. R. Kakivaya, and J. R. Stone (1998). Intractability of the dial-a-ride problem and a multiobjective solution using simulated annealing. *Engineering Optimization* 30, 91–123.

Beaudry, A., G. Laporte, T. Melo, and S. Nickel (2010). Dynamic transportation of patients in hospitals. *OR Spectrum* 32, 77–107.

Berbeglia, G., J.-F. Cordeau, and G. Laporte (2010). Dynamic pickup and delivery problems. *European Journal of Operational Research* 202, 8–15.

Boffey, T. B. (1996). Multiobjective routing problems. *TOP* 3, 167–220.

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

Clímaco, J. C. N., J. M. F. Craveirinha, and M. M. B. Pascoal (2006). An automated reference point-like approach for multicriteria shortest path problems. *Journal of Systems Science and Systems Engineering* 15, 314–329.

Colorni, A. and G. Righini (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. and G. Laporte (2003). A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research Part B* 37, 579–594.

Cordeau, J.-F. and G. Laporte (2007). The dial-a-ride problem : Models and algorithms. *Annals of Operations Research* 153, 29–46.

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

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

Ehrgott, M. (2000). Approximation algorithms for combinatorial multicriteria optimization problems. *International Transactions in Operational Research* 7, 5–31.

Ehrgott, M. and X. Gandibleux (2002). *Multiple Criteria Optimization. State of the art annotated bibliographic surveys*. Kluwer Academic, Dordrecht.

Fonseca, C. M., L. Paquete, and M. Lopez-Ibanez (2006). An improved dimension-sweep algorithm for the hypervolume indicator. In *IEEE Congress on Evolutionary Computation*, Vancouver, pp. 1157–1163.

Gandibleux, X., N. Mezdaoui, and A. Fréville (1997). A tabu search procedure to solve multiobjective combinatorial optimization problems. In R. Caballero, F. Ruiz, and R. Steuer (Eds.), *Advances in Multiple Objectives and Goal Programming*, Volume 455 of *Lecture Notes in Economics and Mathematical Systems*, pp. 291–300. Berlin: Springer.

Gupta, A. and A. Warburton (1986). Approximation methods for multiple criteria travelling salesman problems. In Y. Sawaragi (Ed.), *Towards Interactive and Intelligent Decision Support Systems: Proceedings of the 7. International Conference on Multiple Criteria Decision Making, Kyoto*, Volume 285, pp. 211–217. Berlin: Springer.

Hansen, M. P. (2000). Tabu search for multiobjective combinatorial optimization: Tamoco. *Control and Cybernetics* 29, 799–818.

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

Jørgensen, R. M., J. Larsen, and K. B. Bergvinsdottir (2007). Solving the dial-a-ride problem using genetic algorithms. *Journal of the Operational Research Society* 58, 1321–1331.

Jozefowiez, N., G. Laporte, and F. Semet (2012). A generic branch-and-cut algorithm for multiobjective optimization problems: application to the multilabel traveling salesman problem. *INFORMS Journal on Computing* 24, 554–564.

Jozefowiez, N., F. Semet, and T. El-Ghazali (2008). Multi-objective vehicle routing problems. *European Journal of Operational Research* 189, 293–309.

Karabuk, S. (2009). A nested decomposition approach for solving the paratransit vehicle scheduling problem. *Transportation Research Part B* 43, 448–465.

Lust, T. and A. Jaszkiewicz (2010). Speed-up techniques for solving large-scale biobjective TSP. 37, 521–533.

Lust, T. and J. Teghem (2010). The multiobjective traveling salesman problem: A survey and a new approach. In C. C. Coello, C. Dhaenens, and L. Jourdan (Eds.), *Advances in Multi-Objective Nature Inspired Computing*, Volume 272 of *Studies in Computational Intelligence*, pp. 119–141. Berlin: Springer.

Madsen, O. B. G., H. F. Ravn, and J. M. Rygaard (1995). A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities and multiple objectives. *Annals of Operations Research* 60, 193–208.

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

Paquete, L. and T. Stützle (2003). A two-phase local search for the biobjective traveling salesman problem. In *Evolutionary Multi-Ccriterion Optimization, Proceedings*, Lecture notes in computer science, Berlin, pp. 479–493. Springer.

Paquette, J., F. Bellavance, J.-F. Cordeau, and G. Laporte (2012). Measuring quality of service in dial-a-ride operations: The case of a Canadian city. *Transportation* 39, 539–564.

Parragh, S. N., K. F. Doerner, R. F. Hartl, and X. Gandibleux (2009). A heuristic two-phase solution approach for the multi-objective dial-a-ride problem. *Networks* 54, 227–242.

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

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

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

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

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

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

Wilson, N. H. M., R. W. Weissberg, and J. Hauser (1976). Advanced dial-a-ride algorithms research project: Final report. Working paper R-76-20, Department of Civil Engineering, Massachusetts Institute of Technology, Cambridge, MA.

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

Zitzler, E., L. Thiele, and J. Bader (2010). On set-based multiobjective optimization. *IEEE Transactions on Evolutionary Computation* 14, 58–79.

Zitzler, E., L. Thiele, M. Laumanns, C. M. Fonseca, and G. V. da Fonseca (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation* 7, 117–132.