CIRRELT

Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

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

Julie Paquette
Jean-François Cordeau
Gilbert Laporte
Marta M.B. Pascoal

January 2012

CIRRELT-2012-04

Bureaux de Montréal :
Université de Montréal
C.P. 6128, succ. Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

Université Laval · UQÀM · HEC Montréal · École Polytechnique Montréal · Université de Montréal

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

Julie Paquette[1,2,*], Jean-François Cordeau[1,2], Gilbert Laporte[1,3], Marta M.B. Pascoal[4]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

[2] Department of Logistics and Operations Management, HEC Montréal, 3000 Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

[3] Department of Management Sciences, HEC Montréal, 3000 Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

[4] Departamento de Matematica da Universidade de Coimbra, Apartado 3008, EC Universidade, 3001-454 Coimbra, Portugal and Institute for Systems and Computers Engineering, Coimbra (INESCC), rua Antero de Quental, 199, 3000 – 033 Coimbra, Portugal

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 objective considered in this paper includes three terms related to quality of service. This gives rise to a multicriteria problem. The problem is solved by means of a metaheuristic which efficiently integrates the reference point method for multicriteria optimization within a tabu search mechanism. Extensive tests are 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 be employed to determine good compromises between cost and quality of service.

Keywords. Dial-a-ride problem, multicriteria analysis, quality of service, tabu search, reference point method.

Acknowledgements. This work was partly supported by the Social Sciences and Humanities Research Council of Canada (SSHRC), by the Natural Sciences and Engineering Council of Canada (NSERC) 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.

---

* Corresponding author: Julie.Paquette@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2012

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 variant of the DARP. For recent 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. 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 costs 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. 2011). 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 the 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 an 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 intent 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 manager identify the priorities of the provider, as defined by weights in an objective function. The algorithm can also be used at an operational level, with the weights identified at the tactical level, in order 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 vertex 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 vertex v_i and a destination vertex v_{i+n}. Vertex v_0 represents the depot from which the vehicles start their route, and vertex 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 vertices excluding the vertices associated with the depots, by V'_{OD} the subset of destination vertices associated to outbound trips, and by V'_{IO} the subset of origin vertices associated to inbound trips.

Vehicle k has a capacity of Q_k and performs a route whose duration must not exceed T_k. Vertex $v_i \in V$ has a load equal to q_i, with $q_i = -q_{i+n}$, a service time d_i, and a time window $[e_i, l_i]$. The time window of the depot vertices 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 vertices visited by R_k. We define binary variables x^k_{ij} 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 vertex v_i is denoted by A^k_i, the service time of vehicle k at vertex v_i is equal to $B^k_i \geq \max\{e_i, A^k_i\}$, 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 vertex 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 $L_i^k = B_{i+n}^k - D_i^k$. This variable may not exceed a maximal allowed ride time L. Finally, variable Q_i^k denotes the load of vehicle k after visiting vertex 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:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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 vertex is visited exactly once and that the origin and destination of a request are visited by the same vehicle. The consistency of time variables and vehicle loads 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 time never exceeds L. The maximum length of a route is imposed by constraints (12), and time windows are imposed by constraints (13), (14) and (15).

We have modified the standard DARP 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.

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 Q_1^k ambulatory users and Q_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 Q_1^k and Q_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 after visiting vertex v_i. Constraints (8) and (10) are then duplicated to account for each user type:

$$Q_{1j}^k \ge (Q_{1i}^k + q_{1j})x_{ij}^k \qquad (v_i, v_j \in V', k = 1, ..., m) \tag{17}$$

$$Q_{2j}^k \ge (Q_{2i}^k + q_{2j})x_{ij}^k \qquad (v_i, v_j \in V', k = 1, ..., m) \tag{18}$$

$$\max\{0, q_{1i}\} \le Q_{1i}^k \le \min\{Q_{1k}, Q_{1k} + q_{1i}\} \qquad (v_i \in V', k = 1, ..., m) \tag{19}$$

$$\max\{0, q_{2i}\} \le Q_{2i}^k \le \min\{Q_{2k}, Q_{2k} + q_{2i}\} \qquad (v_i \in V', k = 1, ..., m). \tag{20}$$

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. 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 throught the inclusion of the term

$$a(s) = \sum_{v_i \in V'} \sum_{v_j \in V'} \sum_{k=1}^{m} x_{ij}^k (Q_{1i}^k + Q_{2i}^k) W_j^k \tag{21}$$

in the objective function. The maximal duration of route k is equal to T_k, and drivers are entitled to a 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 $[e_i = b_0 + T_k/2 - 2\nu, l_i = b_0 + T_k/2]$. Another constraint specifies that no user can be onboard the vehicle while the driver takes his break:

$$b(s) = \sum_{v_i \in V'} \sum_{v_j = v_{2n+2}}^{v_{2n+2+m_1}} \sum_{k=1}^{m_1} x_{ij}^k (Q_{1i}^k + Q_{2i}^k) \leq 0. \tag{22}$$

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

$$Cost = \sum_{k=1}^{m_1} c_1 (B_{2n+1}^k - B_0^k)/60 + \sum_{v_i \in V} \sum_{v_j \in V} \sum_{k=m_1+1}^{m} c_2 x_{ij}^k + \sum_{v_i \in V} \sum_{v_j \in V} \sum_{k=m_1+1}^{m} c_3 x_{ij}^k, \tag{23}$$

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. 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:

$$Quality1 = \sum_{v_i \in V'_{OD}} \sum_{k=1}^{m} ((l_i - e_i)/2 - B_i^k)^2. \tag{24}$$

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 quality of the service as being satisfactory. We again use a quadratic inconvenience curve to measure this type of inconvenience:

$$Quality2 = \sum_{v_i \in V'_{IO}} \sum_{k=1}^{m} (B_i^k - e_i)^2. \tag{25}$$

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:

$$Quality3 = \sum_{v_i = v_1}^{v_n} \sum_{k=1}^{m} (L_i^k / t_{i,i+n})^2. \tag{26}$$

These modifications were incorporated into the model to represent the constraints and objectives of the provider under study. It is clear, however, that other constraints or objectives could apply in different contexts.

3 Multicriteria tabu search algorithm

Many 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 iteractions between the decision maker and the optimization phase happen; and *a priori* decision making, when the preferences are aggregated before the solutions are calculated. 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 or branch-and-bound, 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) and in the comprehensive 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 inefficent 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. 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, which 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.

The objective function $f(s)$ minimizes the routing cost $c(s)$ and 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. 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

We now explain the main changes made to this heuristic to handle multiple objectives. 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 ((19) and (20)), 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 (22)). 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 generic multicriteria problems in which the aim is to determine a set of non-dominated feasible solutions which constitute an approximation of the Pareto front. Each solution is represented by a vector whose elements are the values taken by the terms of the objective function. 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 corresponds to a unique non-dominated solution. Figure 1 summarizes these concepts visually. 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 find no indication that the other two metrics yield better solutions. The objective function we have used is defined as the linear combination

$$\begin{aligned} f(s) &= \omega_0|Cost - Cost^*| + \omega_1|Quality1 - Quality1^*| + \omega_2|Quality2 - Quality2^*| \\ &\quad +\omega_3|Quality3 - Quality3^*| + \alpha_1 q_1(s) + \alpha_2 q_2(s) + \beta d(s) + \gamma w(s) + \tau t(s) \\ &\quad +\delta a(s) + \chi b(s), \end{aligned} \tag{27}$$

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

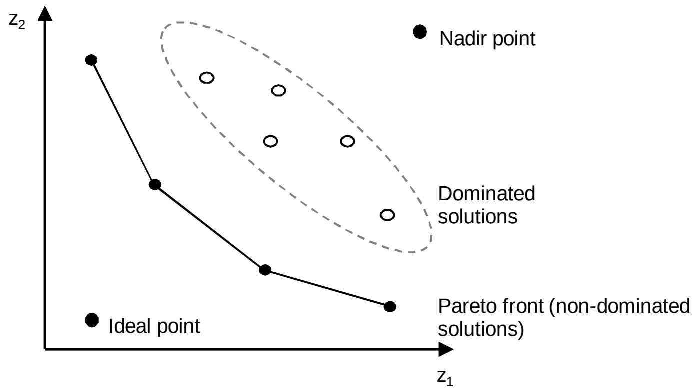

Figure 1: Representation of multicriteria concepts

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, it is necessary to measure all terms of the objective function on a comparable scale. As suggested by Clímaco et al. (2006), we use the 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 feasible solution was encountered, or whenever the algorithm has not identified any feasible solution for more than ζ consecutive iterations. The procedure used to update the weights is 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 must be equal to one. It also allows the weights used at 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 $\omega_2 \geq 0.25$, the forward time slack procedure is not applied. Indeed, this procedure delays as much as possible the beginning of service at the vertices, 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 $p(\bar{s})$. 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

   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 exchange 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 $iterationtotal = 1$.
      (c) If $iterationtotal = \iota$ or if $iterationbest = \zeta$
         i. Adjust the weights $\omega_0, \omega_1, \omega_2, \omega_3$.
         ii. Reset tabu list.
   6. Increment counters *iteration*, *iterationbest* and *iterationtotal*.

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

The random instances are those introduced by Cordeau and Laporte (2003) which have been modified to include two types of users and two types of vehicles. 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. A time window is associated to each origin vertex for an inbound trip and to each destination vertex 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.

Table 1: Characteristics of the random instances

Instance	n	$m1$	$m2$	Instance	n	$m1$	$m2$
1	24	1	2	6	144	5	8
2	48	2	3	7	36	2	2
3	72	3	4	8	72	3	3
4	96	3	6	9	108	3	5
5	120	5	7	10	144	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. The values of Q_1^k and Q_2^k are 8 and 5 for minibuses, 4 and 0 for regular taxis, and 1 and 1 for 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 have a break of $\nu = 30$ minutes, usually around the middle of their working day. The last two constraints are not applied 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.

Table 2: Characteristics of the real-life instances

Instance	Day of the week	n	m_1	m_2	m_3
February 5	Monday	1056	22	20	125
February 6	Tuesday	1136	22	30	130
February 7	Wednesday	1146	22	80	175
February 10	Saturday	454	22	15	50
February 11	Sunday	450	19	5	50
August 6	Monday	854	22	20	125
August 7	Tuesday	970	22	50	125
August 8	Wednesday	986	22	50	150
August 9	Thursday	980	22	60	125
August 10	Friday	882	19	20	125
August 11	Saturday	206	19	5	50
August 12	Sunday	190	19	5	30

4.2 Solution quality indicators

To measure solution quality, 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. 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 set B of solutions is better than the set A. 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 the non-dominated solutions found for the same instance accross all the 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.

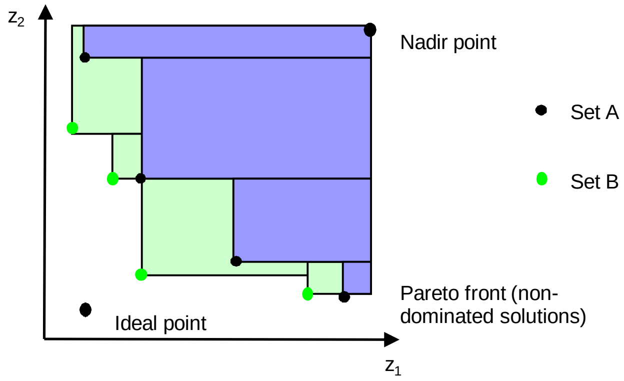

Figure 2: Hypervolume indicator for two sets of solutions

4.3 Preliminary tests on the random instances

Preliminary tests to parametrize the algorithm were executed with the random instances and aimed to

1. identify the best values for ι and ζ, the numbers of iterations after which the weights are changed,
2. 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, and
3. determine whether the regular diversification procedure is useful in the multicriteria context.

Each random instance was solved ten times with ten different seeds for 10^6 iterations. The algorithm was tested for the following values of (ι, ζ): (100, 500), (100, 1000), (100, 50000), (1000, 5000), (1000, 50000), and (10000, 50000). Three values were tested for δ: 0.5, 1, and 1.5. The algorithm was also tested with and without the regular diversification procedure. Thus, 36 parametrization combinations were tested. The values of the lower and upper bounds for each optimization criterion are summarized in Table 3. These values are an approximation of the best values found when individually optimizing each objective term five times.

Table 4 presents the average of the hypervolume and epsilon indicators on all instances for the different parametrizations based on ι and ζ values. According to the hypervolume indicator, the best parametrization is the one with $\iota = 1000$ and $\zeta = 50000$. However, according to

Table 3: 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

the epsilon indicator, the best parametrization would be $\iota = 10000$ and $\zeta = 50000$. Thus the values for ι and ζ should be fixed to one of the largest value possible considering the maximum number of iterations allowed.

Table 5 presents aggregated results over the values of δ. The best value is 0.5 according to both indicators. Finally, Table 6 shows that it is preferable to apply the regular diversification procedure (Cordeau and Laporte 2003) within our multicriteria algorithm, even though other factors act as diversification agents.

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

ι	ζ	Hypervolume indicator	Epsilon indicator
100	500	1.42246E+17	8.03
100	5000	1.43666E+17	8.86
100	50000	1.51939E+17	7.25
1000	5000	1.41795E+17	10.55
1000	50000	1.52557E+17	6.87
10000	50000	1.49727E+17	5.05

Table 5: 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	1.77318E+17	5.15
1	1.60744E+17	5.84
5	1.02979E+17	12.40

Table 6: Indicator values according to the fact that the diversification procedure is used or not (average on all instances)

Diversification	Hypervolume indicator	Epsilon indicator
No	1.43239E+17	9.15
Yes	1.50731E+17	6.38

4.4 Application to the real-life instances

Having parametrized the algorithm on the random instances, we have applied it with the best parameter setting to the real-life instances. 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 7 and 8). We have used the parameter values derived from our tests on the random instances, except for ι and ζ which depend on the total number of iterations. Because 10^5 iterations were necessary to solve the real-life instances, we set (ι, ζ) to (100, 1000) and (100, 500). Table 9 presents for each instance the total computing time (minutes), the number of solutions found and the average computing time (minutes) per non-dominated solution. The best setting is (100, 1000), which is consistent with the results obtained on the random instances.

Table 7: Upper and lower bounds for real-life instances from Monday to Friday

$\underline{Cost}$	$\underline{Quality1}$	$\underline{Quality2}$	$\underline{Quality3}$	$\overline{Cost}$	$\overline{Quality1}$	$\overline{Quality2}$	$\overline{Quality3}$
10000	0	98000	0	30000	60000	300000	20000

Table 8: Upper and lower bounds for real-life instances on Saturday and Sunday

$\underline{Cost}$	$\underline{Quality1}$	$\underline{Quality2}$	$\underline{Quality3}$	$\overline{Cost}$	$\overline{Quality1}$	$\overline{Quality2}$	$\overline{Quality3}$
0	0	20000	0	20000	20000	100000	10000

Table 9: Computational results for the multicriteria algorithm on the real-life instances. (1) Execution with $(\iota, \zeta) = (100, 500)$; (2) Execution with $(\iota, \zeta) = (100, 1000)$

Instance	Computing time in minutes (1)	Number of solutions found (1)	Computing time in minutes per non-dominated solution (1)	Computing time in minutes (2)	Number of solutions found (2)	Computing time in minutes per non-dominated solution (2)
February 5	1360.43	107	12.71	1361.25	107	12.72
February 6	1323.33	7	189.05	1538.96	5	307.79
February 7	941.24	194	4.85	807.85	4	201.96
February 10	196.4	23	8.54	326.06	100	3.26
February 11	45.54	24	1.90	32.75	4	8.19
August 6	251.33	50	5.03	247.05	50	4.94
August 7	235.85	51	4.62	228.45	6	38.08
August 8	606.79	70	8.67	606.51	70	8.66
August 9	396.87	126	3.15	394.08	126	3.13
August 10	380.48	8	47.56	314.28	37	8.49
August 11	17.25	19	0.91	16.67	5	3.33
August 12	18.92	48	0.39	15.4	34	0.45
Average	**481.20**	**60.58**	**23.95**	**490.78**	**45.67**	**50.08**

The average computing time is 480 minutes, which is acceptable considering that in practice, the multicriteria problem will only be solved infrequently at the tactical level. Indeed, the trade-off between costs and the quality criteria is only considered periodically. For the purpose of daily planning, the weights can be set equal to the best ones identified in the last tactical analysis performed.

4.5 Trade-offs between cost and quality

The solution of real-life instances using the multicriteria tabu search heuristic can also help assess the trade-offs between cost and quality of service. To this end, we have performed tests on six of the real-life instances: February 5, February 10, February 11, August 8, August 11, and August 12. For each of the quality criteria, each of these instances was solved with four different weight combinations. For $Quality1$, we have successively fixed ω_1 to 0.2, 0.4, 0.6, and 0.8 and we have set ω_0, ω_2, and ω_3 equal to $(1 - \omega_1)/3$. We have proceeded in a similar fashion for $Quality2$ and $Quality3$ by varying the value of ω_2 and ω_3. The tabu search algorithm was again executed in a single-thread fashion, but instead of modifying the weights dynamically, we have successively used each of the four settings just defined.

Having solved an instance with a given set of weights, we have averaged the values of $Cost$, $Quality1$, $Quality2$, and $Quality3$ over all non-dominated solutions identified by the algorithm. For example, the weights and average solution values corresponding to a variation of ω_1 on the February 11 instance are reported in Table 10 and depicted in Figure 3. In this case, it can be observed that increasing ω_1 from 0.2 to 0.8 decreases the inconvenience associated with $Quality1$ by 33.45% but increases $Cost$ by only 10.81%. Here, the improvement in $Quality1$ is mostly achieved at the expense of $Quality2$, without any noticeable effect on $Quality3$.

Table 10: Values of $Cost$, $Quality1$, $Quality2$, and $Quality3$ with the four different weight combinations for the February 11 instance associated with a variation of ω_1

Weight combination	$Cost$	$Quality1$	$Quality2$	$Quality3$
(0.27; 0.2; 0.27; 0.27)	5170.08	2263.13	499.55	642.30
(0.2; 0.4; 0.2; 0.2)	5500.57	1928.40	1184.49	561.21
(0.13; 0.6; 0.13; 0.13)	5666.75	1831.10	1361.78	580.87
(0.07; 0.8; 0.07; 0.07)	5728.91	1506.11	2866.57	661.80

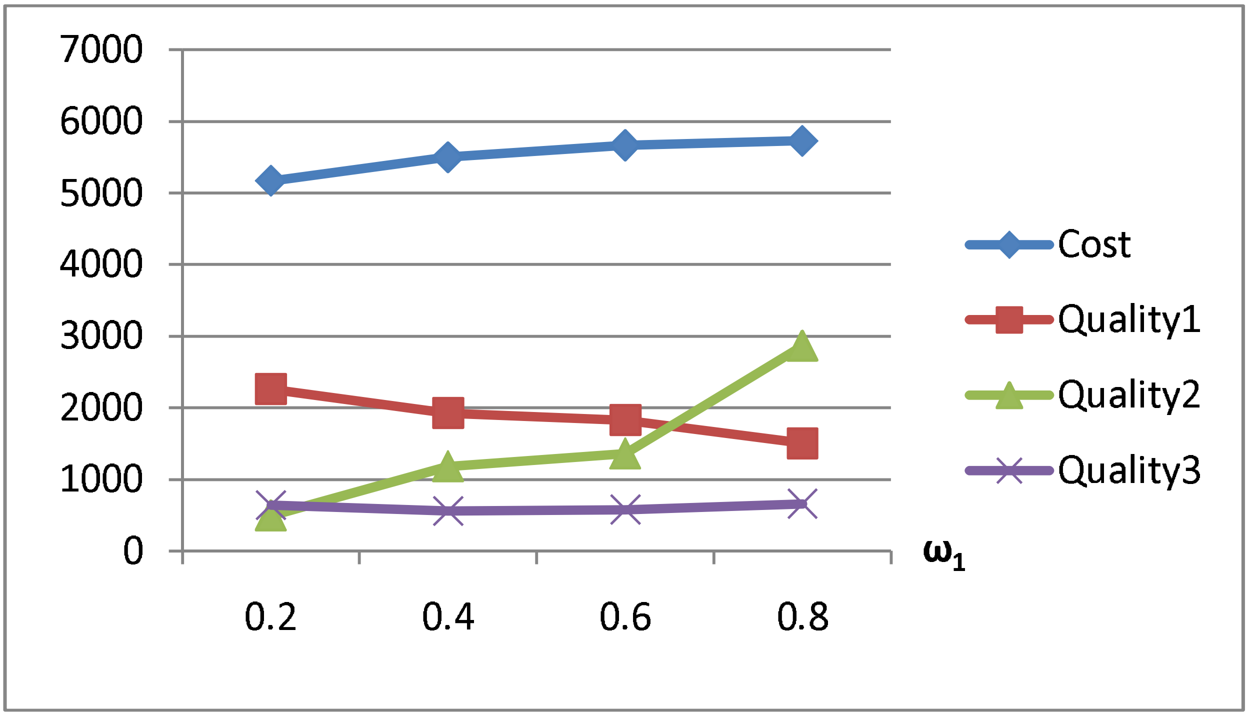

Figure 3: Values of $Cost$, $Quality1$, $Quality2$, and $Quality3$ for the February 11 instance associated with a variation of ω_1

Similar results are reported in Tables 11, 12 and 13 for all six instances and each quality criterion. In each case, significant average inconvenience reductions (51.82%, 83.85% and 38.91%) can be obtained by increasing the solution cost by only a small percentage (7.21%, 5.75% and 0.33%). Increasing ω_2 has a relatively small effect on *Cost* and *Quality1* but increases the inconvenience related to *Quality3* rather significantly. Increasing ω_3 yields marginally higher *Cost* values, but much worse solutions in terms of *Quality1* and *Quality2*. This type of analysis can prove highly useful to dial-a-ride operators who seek cost-efficient compromises between solution cost and quality of service, or between several quality attributes.

Table 11: Variations of the cost and the three quality of service criteria on six real-life instances when more emphasis is put on *Quality1*

Real-life instance	Inconvenience variation linked to *Quality1*	*Cost* variation	Inconvenience variation linked to *Quality2*	Inconvenience variation linked to *Quality3*
February 5	−60.02%	−0.45%	152.62%	43.94%
February 10	−63.74%	14.05%	432.34%	67.24%
February 11	−33.45%	10.81%	473.83%	3.03%
August 8	−64.45%	13.29%	24.83%	22.37%
August 11	−44.82%	0.50%	659.32%	23.88%
August 12	−44.44%	5.08%	167.37%	43.92%
Average	**−51.82%**	**7.21%**	**318.39%**	**34.06%**

Table 12: Variations of the cost and the three quality of service criteria on six real-life instances when more emphasis is put on *Quality2*

Real-life instance	Inconvenience variation linked to *Quality2*	*Cost* variation	Inconvenience variation linked to *Quality1*	Inconvenience variation linked to *Quality3*
February 5	−76.50%	−0.41%	41.06%	31.11%
February 10	−68.49%	7.94%	−68.49%	2.88%
February 11	−96.23%	11.95%	−35.64%	24.05%
August 8	−63.31%	7.41%	26.71%	19.75%
August 11	−99.02%	7.00%	−5.62%	12.00%
August 12	−99.53%	0.61%	41.71%	37.59%
Average	**−83.85%**	**5.75%**	**−0.04%**	**21.23%**

Table 13: Variations of the cost and the three quality of service criteria on six real-life instances when more emphasis is put on *Quality3*

Real-life instance	Inconvenience variation linked to *Quality3*	*Cost* variation	Inconvenience variation linked to *Quality1*	Inconvenience variation linked to *Quality2*
February 5	−45.89%	2.24%	41.38%	−31.46%
February 10	−38.92%	−7.21%	27.59%	−21.26%
February 11	−34.51%	2.48%	8.52%	16.35%
August 8	−20.83%	3.03%	25.55%	29.37%
August 11	−45.40%	−0.20%	10.13%	61.05%
August 12	−47.97%	1.64%	−0.26%	2.82%
Average	**−38.91%**	**0.33%**	**18.82%**	**9.48%**

5 Conclusions

We have developed a multicriteria heuristic embedding a tabu search process to the solution of DARPs combining cost and quality of service criteria. Our algorithm is the first to 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 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.

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. Springer Berlin / Heidelberg.

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 Objective and Goal Programming*, Volume 455 of *Lecture Notes in Economics and Mathematical Systems*, pp. 291–300. Berlin: Springer Berlin / Heidelberg.

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, Berlin, pp. 211–217. Springer Verlag.

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., 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. Springer Berlin / Heidelberg.

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 *EMO*, pp. 479–493.

Paquette, J., F. Bellavance, J.-F. Cordeau, and G. Laporte (2011). Measuring quality of service in dial-a-ride operations: The case of a Canadian city. Working paper CIRRETL-2011-19, CIRRELT, Montréal, Canada.

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.