A Multiobjective Model and Simulated Annealing Approach for a Dial-a-Ride Problem

G.R. Mauri[1,2] and L.A.N. Lorena[2]
[1]Rural Engineering Department - Center for Agrarian Science - CCA
Federal University of Espírito Santo - UFES
Alegre - ES - BRAZIL
[2]Laboratory for Computing and Applied Mathematics - LAC
National Institute for Space Research - INPE
São José dos Campos - SP - BRAZIL
E-mail: mauri@cca.ufes.br, lorena@lac.inpe.br

Abstract

This paper describes a general multiobjective mathematical model for a dial-a-ride problem and an application of Simulated Annealing to solve it. The model deals with a static mode of the problem and comprehend several distinct cases of the regular problem such as heterogeneous or homogeneous fleet of vehicles, multi or single depot and a multiobjective minimizing function that treats transportation costs and customer inconveniences over weighting against each other. The Simulated Annealing application is simple, but for the new neighbors' generation, three types of moves are randomly used through a uniform distribution, and the routes are clustered and scheduled in a separate way of other heuristic methods. Computational results are performed over publicly available data sets and the results are compared against current state-of-the-art methods.

Keywords: dial-a-ride problem, simulated annealing, multiobjective model.

1. INTRODUCTION

In general, the Dial-a-Ride Problem (DARP) shares several features with pick-up and delivery problems arising in courier services. However, since it is concerned with transporting people, the level of service criteria (the "quality" of service) becomes more important and complex. Thus, punctuality, reduction of the waiting time and routes duration are more critical in DARP than in other similar problems (Cordeau & Laporte, 2003a).

DARP is a known NP-Hard problem, and several methodologies have been adopted to solve it. However, as shown in Cordeau (2004), an exact mathematical formulation to solve it is very complex, and unable of being solved in a reasonable time for problems of "real size".

The use of heuristics and metaheuristics to solve DARP is increasing significantly, because such methods, in spite of not guarantee optimal solutions, such allow inserting several constraints in a soft way. Among these methods, metaheuristics as Genetic Algorithms, Tabu Search, Simulated Annealing, among others, admit to easily include several types of conditions that treat the customers' satisfaction and the operational costs simultaneously (see Baugh jr. *et al* (1998), Bergvinsdottir (2004), Bergvinsdottir *et al* (2004), Hart (1996), Cordeau & Laporte (2003b) and Znamensky & Cunha (1999)). Besides, several specific heuristics also exist for solve this problem (see Jaw *et al* (1986), Madsen *et al* (1995), Toth & Vigo (1996) and Toth & Vigo (1997)).

This work presents a simple and effective alternative to solve a dial-a-ride problem. A general mathematical and multiobjective model is proposed to represent the problem, and the metaheuristic Simulated Annealing is used with other heuristics to treat it, to produce routes to be operationally economical and satisfying the customers' demand keeping a "good" quality level in the realized service.

The paper is organized as follows. Section 2 presents a brief description of the dial-a-ride problem. The model is detailed in Section 3, while Section 4 describes the methods used to solve it. The computational results are presented in Section 5, and the conclusions are summarized in Section 6.

2. PROBLEM DESCRIPTION

Considering a classification of vehicle routing and programming problems, those that cover the passengers' transport of their origin places to their destinies are known in literature as problems of “dial-a-ride” type (Znamensky & Cunha, 1999).

Dial-a-Ride - DARP (see Cordeau (2004), Cordeau & Laporte (2003a) and Cordeau & Laporte (2003b)) consists of developing routes and scales of vehicles to transport several customers, which specify requests of pick-up and delivery between specific origin and destiny places. The objective of that process is planning a set of routes for some vehicles, with “minimal cost”, holding the largest possible number of customers, always complying a set of constraints.

In deficient people transportation, a pick-up point matches to the address where a certain customer should get into the vehicle, and the associated delivery point matches to the address where the same customer should leave the vehicle. Each pick-up point and its respective delivery point and their respective “time window” define a customer’s *Transportation Request*.

A common tendency in DARP models is to let that customers fix a “time window” (time intervals for their attendance) for their departure and arrival, because according to Jaw *et al* (1986), the customers should be able to specify a time interval for their pick-up and delivery, both in specific places.

Each vehicle has a capacity, usually measured in number of conventional seats and number of wheelchairs, for instance. Similarly, for each transportation request is associated an occupation amount of conventional seats (eventual companions) and wheelchairs (Znamensky & Cunha, 1999).

According to Cordeau & Laporte (2003a), and other several authors, DARP may operate according to a static or dynamic mode. In the first case, all the transportation requests are known in advance, while in second case the requests are done during the day (usually by telephone), and the routes of the vehicles are adjusted in real time agreeing with the demand. However, in practice, rarely “pure” dynamic DARPs exist, because usually a subset of requests is known in advance.

Most of the studies about DARP assumes the availability of a set of homogeneous vehicles installed in a single depot, however, it is important to notice that in practice different situations exist, as for instance: may exist several depots, especially in great geographical areas, and sometimes the set of vehicles is heterogeneous, being some vehicles projected for only transport wheelchairs, other for only transport passengers for clinics and still some able of holding both passenger types (Cordeau & Laporte, 2003a).

DARP can be considered with several different objectives: minimizing operational costs subject to satisfaction of whole demand; maximizing demand’s satisfaction, subject to availability of vehicles, or other combinations of these, that usually look for a balance between operational cost and quality of offered service. Usually, the evaluation criterion used in DARP resolution includes the total distance traveled by the vehicles, the duration of routes, the average waiting time of vehicles in the pick-up and delivery places and the average ride time of the customers (time that customers are inside the vehicles), and to each one of those is given a larger or smaller importance to the other ones.

In general terms, DARP generalizes several problems of vehicles routing as the Pick-up and Delivery Vehicle Routing Problem (PDVRP) and Vehicle Routing Problem with Time Windows (VRPTW). DARP is different from most of such problems because of its human perspective, in the passengers' transportation, reducing the customers' “inconvenience” (quality of service) that should be balanced with the minimization of operational costs (Calvo *et al*, 2004, Cordeau & Laporte, 2003a and Xu *et al*, 2003).

3. PROPOSED MODELING

The model proposed in this work treats DARP in a static mode (all the requests are known in advance), with multiple vehicles, with heterogeneous fleet (each vehicle has a different capacity) and multiple depots (each vehicle begins and finishes its route in specific depots). This is an approach that includes several others found in literature (see Bergvinsdottir (2004), Bergvinsdottir *et al* (2004), Cordeau & Laporte (2003a) and Cordeau & Laporte (2003b)), and it is very close to real situations.

In this model, the existence of n customers (transportation requests) is initially assumed to be assisted by m vehicles. Each transportation request specifies a place of pick-up i and one of delivery $n+i$. To represent the problem, the following sets are defined:

- K: set of available vehicles ($|K| = m$).
- G^-: set of origin depots.
- G^+: set of destiny depots.
- P: set of pick-up places.
- U: set of delivery places.
- $N = G^- \cup P \cup U \cup G^+$: set of all places (points).

Each customer i ($\forall\ i \in P$) specifies the "load" necessary q_i for his transportation, in other words, the number of seats in vehicle that he will occupy, and two time intervals, one in that he would like to be picked-up in his origin $[e_i,l_i]$, and another that he would like to be delivered in his destiny $[e_{n+i},l_{n+i}]$ (time windows). The load q_i should be a positive value in the pick-up places and the same value, however negative, in the respective delivery places.

Each vehicle k ($\forall\ k \in K$) has a known capacity Q''_k (amount of available seats), a maximum time of duration associated to its route T''_k, and it begins its route in a specific depot g^-_k and finishes in another g^+_k (the same or not). All the depots still have their own time windows for departure (origin depot) and arrival (destiny depot).

To each customer i ($\forall\ i \in P$) is associated a maximum ride time R''_i, in other words, the maximum time that customer will be able to be inside the vehicle. To each place i ($\forall\ i \in \{P \cup U\}$) is associated a maximum waiting time W''_i, which is the maximum time that vehicles can wait until beginning the "service". It is considered as service the pick-up or delivery of a customer in a certain place, assigning a time s_i necessary for its finish. Finally, with the places about depots and the points of transportation requests, the distances $d_{i,j}$ and the travel time $t_{i,j}$ between the points i and j, $\forall\ i,j \in$ N and $i \neq j$ are given.

Then, the objective is to minimize the operational costs and the customers' "inconvenience", that is, minimizing the "non-essential" requirements of problem. These requirements are related to the total distance traveled by the vehicles, to the number of vehicles used in problem solution, to the routes duration, to customers' ride time and to waiting times in pick-up and delivery places.

A set of essential requirements should be still complied. These requirements should be obligatorily assisted for getting a valid solution (feasible in practical terms) to the problem. These requirements are:

- The duration of a route performed by the vehicle k ($\forall\ k \in K$) should not exceed the allowed maximum time T''_k.
- The ride time of customer i ($\forall\ i \in P$) should not exceed the maximum ride time allowed R''_i;
- The waiting time in place i ($\forall\ i \in \{P \cup U\}$) should not exceed the maximum waiting time allowed W''_i;
- The capacity Q''_k ($\forall\ k \in K$) of the vehicles cannot be exceeded at any place;
- The start of service in all the places i ($\forall\ i \in N$) should be inside the intervals $[e_i,l_i]$.

Using these information, the decision variables that will supply all the vehicles programming can be defined. A_i is considered as arrival time in place i ($\forall\ i \in N$) by the vehicle that will attend it, and $A_i = 0$ if $i \in G^-$ and $A_i = D_{i-1} + t_{i-1,i}$ if $i \in \{P \cup U \cup G^+\}$; D_i is the departure time of place i ($\forall\ i \in N$), again by the vehicle that will attend it, and $D_i = 0$ if $i \in G^+$, $D_i = B_i + s_i$ if $i \in \{P \cup U\}$ and $D_i = B_i$ if $i \in \{G^-\}$; B_i represents the start time of service in place i ($\forall\ i \in N$), and $B_i = D_i$ if $i \in G^-$ and $B_i = \max\{e_i,A_i\}$ if $i \in \{P \cup U \cup G^+\}$; the waiting time before service beginning in place i ($\forall\ i \in N$) is W_i, and $W_i = 0$ if $i \in G^-$ and $W_i = B_i - A_i$ if $i \in \{P \cup U \cup G^+\}$; Q_i indicates the load (number of busy seats) of the vehicle that attends the place i ($\forall\ i \in N$), after service conclusion, and $Q_i = 0$ if $i \in \{G^- \cup G^+\}$ and $Q_i = Q_{i-1} + q_i$ if $i \in \{P \cup U\}$; R_i is the ride time of customer i ($\forall\ i \in P$), and $R_i = B_{n+i} - D_i$.

Finally, $x^k_{i,j} = 1$ if the vehicle k travels from the place i to the place j and $x^k_{i,j} = 0$ otherwise, and the mathematical model is given as follows:

Minimize:

$$\omega_0 \sum_{k\in K}\sum_{i\in N}\sum_{j\in N; j\neq i}(d_{i,j}x^k_{i,j}) + \omega_1 \sum_{k\in K}\sum_{j\in P} x^k_{g^-_k,j} + \omega_2 \sum_{k\in K}(B_{g^+_k} - D_{g^-_k}) + \omega_3 \sum_{i\in P} R_i + \omega_4 \sum_{i\in\{P\cup U\}} W_i + \quad (1)$$

$$\beta_0 \sum_{k\in K}\max\{0,(B_{g^+_k} - D_{g^-_k}) - T''_k\} + \beta_1 \sum_{i\in P}\max\{0, R_i - R''_i\} + \beta_2 \sum_{i\in\{P\cup U\}}\max\{0, W_i - W''_i\} + \quad (2)$$

$$\beta_3 \sum_{k\in K}\max\{0,\left(Q_i \sum_{i\in\{P\cup U\}}\ \sum_{j\in\{P\cup U\}; j\neq i; j\neq n-i} x^k_{i,j}\right) - \tilde{Q}_k\} + \beta_4 \sum_{i\in N}\left(\max\{0, e_i - B_i\} + \max\{0, B_i - l_i\}\right) \quad (3)$$

Subject to:

$$\sum_{j\in\{P\cup\{g^+_k\}\}} x^k_{g^-_k,j} = 1 \qquad \forall k \in K \quad (4)$$

$$\sum_{i\in\{U\cup\{g^-_k\}\}} x^k_{i,g^+_k} = 1 \qquad \forall k \in K \quad (5)$$

$$\sum_{k\in K}\ \sum_{j\in\{P\cup U\};\, j\neq i} x^k_{i,j} = 1 \qquad \forall i \in P \quad (6)$$

$$\sum_{j\in\{P\cup U\};\, j\neq i} x^k_{i,j} - \sum_{j\in\{P\cup U\cup\{g^+_k\}\};\, j\neq i; j\neq n+i} x^k_{n+i,j} = 0 \qquad \forall k \in K; i \in P \quad (7)$$

$$\sum_{j\in\{P\cup U\cup\{g^-_k\}\};\, j\neq i; j\neq n+i} x^k_{j,i} - \sum_{j\in\{P\cup U\};\, j\neq i} x^k_{i,j} = 0 \qquad \forall k \in K; i \in P \quad (8)$$

$$\sum_{j\in\{P\cup U\};\, j\neq i} x^k_{j,i} - \sum_{j\in\{P\cup U\cup\{g^+_k\}\};\, j\neq i; j\neq n-i} x^k_{i,j} = 0 \qquad \forall k \in K; i \in U \quad (9)$$

$$B_j = (B_i + s_i + t_{i,j} + W_j)\sum_{k\in K} x^k_{i,j} \qquad \forall i, j \in N; i \neq j \quad (10)$$

$$Q_j = (Q_i + q_j)\sum_{k\in K} x^k_{i,j} \qquad \forall i, j \in N; i \neq j \quad (11)$$

$$A_i = B_i - W_i \qquad \forall i \in \{P \cup U \cup G^+\} \quad (12)$$

$$D_i = B_i + s_i \qquad \forall i \in \{P \cup U \cup G^-\} \quad (13)$$

$$R_i = B_{n+i} - D_i \qquad \forall i \in P \quad (14)$$

$$A_{g^-_k} = D_{g^+_k} = Q_{g^-_k} = Q_{g^+_k} = W_{g^-_k} = 0 \qquad \forall k \in K \quad (15)$$

$$A_i, W_i, B_i, D_i, Q_i \quad \text{unrestricted} \qquad \forall i \in N \quad (16)$$

$$\text{R}_{\text{i}} \quad \text{unrestricted} \qquad \forall i \in P \tag{17}$$

$$\text{x}_{\text{i,j}}^{\text{k}} \in \{0,1\} \qquad \forall k \in K; \forall i, j \in N; i \neq j \tag{18}$$

The objective function is divided in two parts (equations (1) and equations (2) and (3)). The first part (1) seeks to minimize non-essential requirements of problem, while the second one (2) and (3) seeks to minimize the violations in essential requirements. Besides, the non-essentials requirements are "penalized" through a vector of positive integers numbers (weights) $\boldsymbol{\omega} = [\omega_0,\omega_1,\omega_2,\omega_3,\omega_4]$, and the essentials with a similar vector $\boldsymbol{\beta} = [\beta_0,\beta_1,\beta_2,\beta_3,\beta_4]$.

Constraints (4) and (5) guarantee, respectively, that each vehicle will leave its origin depot and will arrive at its destiny depot once, or in other words, each route will be begun at origin depot and will finish at its destiny depot. The constraint (6) guarantees that each customer will be assisted once, for just a vehicle. The constraint (7) guarantees that a delivery place will always be in the same route that its respective pick-up place, while the flow contention (everything that enters is the same to everything that leaves) is guaranteed by constraints (8) and (9). Constraint (10) settles the time at the beginning of service and the waiting time in each place and the vehicle that will assist it. As shown in Cordeau (2004), these constraints, together, guarantee the elimination of sub-tours (in case of DARP). The constraint (11) determines the load of the vehicles in each place, while the constraints (12), (13) and (14) guarantee, respectively, a correct computation of the arrival and departure times in places and ride times of customers. The constraint (15) "initializes" some variables about the depots. Finally, the constraint (18) just guarantees that decision variables $x^k_{i,j}$ will be binary, while the constraints (16) and (17) guarantee that other variables will be unrestricted.

4. SIMULATED ANNEALING APPLIED TO DARP

Simulated Annealing - SA is a local search method that accepts worsening moves to escape of local optimal. It was proposed originally by Kirkpatrick *et al* (1983), and it is based in an analogy with thermodynamics, when simulating the cooling of a set of heated atoms.

For use SA, a method for generation of an initial solution *S*, a method for generation of neighboring solutions *S'* (neighborhood structure), and an objective function *f(S)* to be optimized should be defined.

4.1. Initial Solution

In this work, the initial solution is formed through a *distribution heuristic*, that is responsible for routing the vehicles, for forming clusters of places in the routes and determine their attending sequence of these. This heuristic is presented in Figure 1.

```
1.  CREATE (m empty routes, and assign it to m vehicles);
2.  CREATE (a list L with all customers' transportation requests);
3.  FOR (each route k, k = 1,2,...,m) DO
4.    SELECT (⌊n/m⌋ requests of L);
5.    FOR (each selected request) DO
6.      Pos1 ← any position of route k;
7.      Pos2 ← any position of route k, but after to Pos1;
8.      INSERT (the pick-up point in Pos1);
9.      INSERT (the delivery point in Pos2);
10.   END-FOR;
11.   INSERT (the origin depot of vehicle k in beginning of route k);
12.   INSERT (the destiny depot of vehicle k at the end of route k);
13.   REMOVE (the requests selected of L);
14. END-FOR;
```

Figure 1. Distribution Heuristic.

In the distribution heuristic, *m* empty routes are created, assigning to each one of them a specific vehicle. Later, all the customers' transportation requests (pick-up points and its respective delivery ones) are randomly distributed in a uniform mode to these routes, and the *n* transportation requests are divided equally among the *m* vehicles. Obviously, sometimes divides *n* by *m* won't be an integer, and in these cases the last route should attend a higher number of requests. Selecting requests and routes that will assist them is also random.

The insertion positions in route are also selected in a random mode, but always assisting the precedence constraint originating from the customers' transportation request, (the pick-up place should always be before the delivery place). Finally, due to the fact that all routes begin and finishes at specific depots (regarding the vehicle that will perform it), the places regarding these depots will always be allocated on first and last points in each one of the created routes.

$B_0 \leftarrow e_0$; $D_0 \leftarrow B_0$;
<u>COMPUTE</u> (A_i, B_i, W_i, D_i, Q_i for each point $v_i \in V_k$ and $v_i \neq v_0$);
<u>COMPUTE</u> (F_0);
$B_0 \leftarrow e_0 + \min\left\{F_0, \sum_{0<p\leq z} W_p\right\}$; $D_0 \leftarrow B_0$;
<u>UPDATE</u> (A_i, B_i, W_i, D_i for each point $v_i \in V_k$ and $v_i \neq v_0$);
<u>COMPUTE</u> (R_i for each point $v_i \in V_k$ and $v_i \in P$);
<u>FOR</u> (each point $v_i \in V_k$ and $v_i \in P$) <u>DO</u>
   <u>COMPUTE</u> (F_i);
   $B_i \leftarrow B_i + \min\left\{F_i, \sum_{i<p\leq z} W_p\right\}$;
   $D_i \leftarrow B_i + s_i$; $W_i \leftarrow B_i - A_i$;
   <u>UPDATE</u> (A_j, B_j, W_j, D_j for each point $v_j \in V_k$ and v_j after to v_i);
   <u>UPDATE</u> (R_j for each point $v_j \in V_k$, $v_j \in P$ and v_{n+j} after to v_i);
<u>END-FOR</u>;

Figure 2. Programming Heuristic.

The distribution heuristic just treats the vehicles' routing, but the programming of these vehicles should still be made to determine the arrival times in places, the departure times, and so on. Then, another heuristic, denominated *programming heuristic* (Figure 2), is used. The programming heuristic is adapted of the one presented in Cordeau & Laporte (2003b), and performs the programming trying to reduce the violations in time windows, in routes duration and in ride times.

A concept of "delay" is used in this heuristic. This concept was initially proposed by Savelsbergh (1992), and consists basically of delaying, as much as possible, the departure time of the origin depot and the beginning of service in pick-up places.

Initially, the departure time of origin depot is set for the start time of the respective time window. So, the other calculations (arrival time, start of service, waiting time, departure time and load of vehicle) are performed for all the next points in route. Later, the delay is computed for departure of origin depot, and then the departure time is adjusted without increase the violations in time windows. Then, it is made an updating in the times for all the next points to origin depot, and it is also computed the customers' ride time. Finally, for each pick-up point in route, its respective delay is computed (and the beginning of service adjusted reducing the route duration and the customers' ride time and without increase the violations in time windows), and the times of all the next points are updated, as well as customers' ride time whose delivery is after the pick-up point in case.

Considering then a route $V_k = \{v_0, v_i, v_j, v_{n+j}, v_{n+i}, \ldots, v_z\}$ performed by the vehicle k ($\forall\ k \in K$), v_0 and v_z can be defined, respectively, by the origin and destiny depots of k ($v_0 \in G^-$; $v_z \in G^+$), and the other points represent the pick-up and delivery points. The points v_{n+i} represents the delivery points corresponding to the pick-up points v_i ($v_i \in P$ and $v_{n+i} \in U$). For any route V_k, the delay is computed as presented in equations (19), (20) and (21), and its programming is shown in Figure 2.

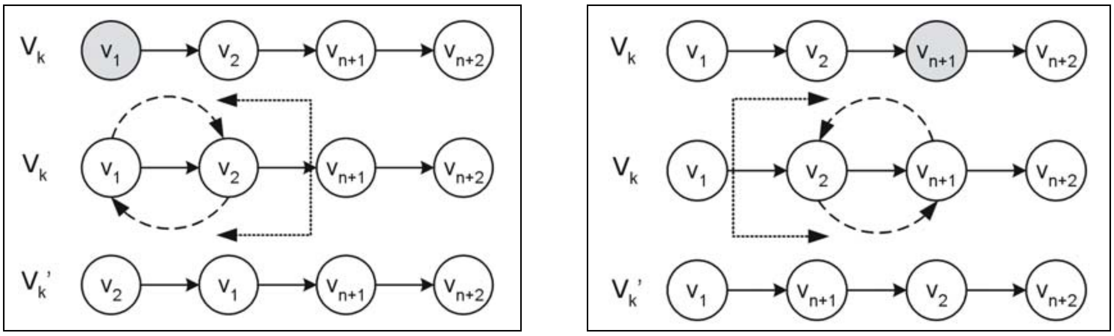
$$F_i = \min_{i \le j \le z} \left\{ \sum_{i<p\le j} W_p + \left(\min\{ l_j - B_j, \tilde{R}_i - R_j \} \right)^{+} \right\} \qquad \text{if } i \in P \tag{19}$$

$$F_i = \min_{i \le j \le z} \left\{ \sum_{i<p\le j} W_p + \left(l_j - B_j \right) \right\} \qquad \text{if } i \in G^{-} \tag{20}$$

$$F_i = 0 \qquad \text{if } i \in \{U \cup G^{+}\} \tag{21}$$

4.2. Neighborhood Structure

Three different change moves were used as neighborhood structure: *Re-order route*, *Re-allocate points* and *Change points*. These moves are based in others found often in works about DARP (see Bergvinsdottir (2004), Bergvinsdottir *et al* (2004), Cordeau & Laporte (2003b), Savelsbergh (1992) and Hart (1996)).



(a) **(b)**

Figure 3. Re-order Route.

It is interesting to highlight that in these moves the depots are not considered, because they are "fixed" in all the routes, and thus their positions cannot be changed.

The Re-order route move consists basically in selecting any route from solution, select any point in this route, select a new position for this point and change this point position with the new position. This move is shown in Figure 3. The selected point can be a pick-up or a delivery point. In first case (Figure 3a), the new position will be, obligatorily, **earlier** to its respective delivery point. In second case (Figure 3b), the new position should be **after** the respective pick-up point. These "limits" are presented through the stippled lines in Figure 3.

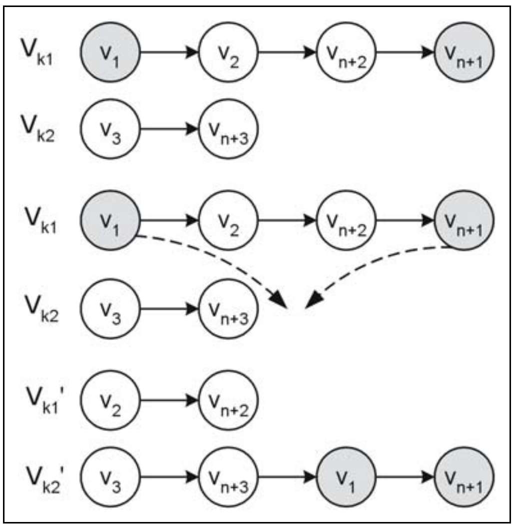


Figure 4. Re-allocate Points.

The Re-allocate points move consists basically in also select two routes from solution, select any transportation request in just one of the two routes, extract it (its pick-up and delivery points) of origin route and add it to the another route, in any position. This move is shown in Figure 4. The pick-up point and its respective delivery point are extracted simultaneously, however its insertion in "another" route can be done in a separate way, in other words, these points are allocated individually in any position of the route, however always keeping the condition that the pick-up point is before the delivery one (precedence constraint).

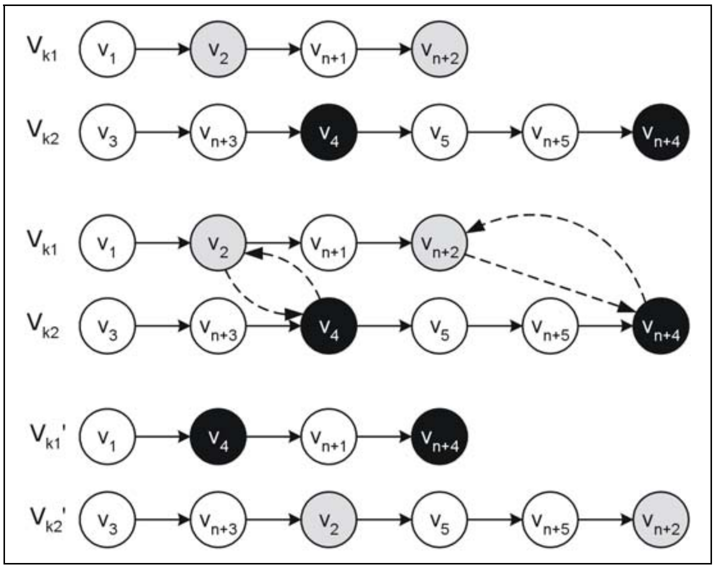


Figure 5. Change Points.

The Change points move consists of selecting any two routes from solution, select any transportation request (pick-up point and its respective delivery) in each one of two routes, and change them. This move is shown in Figure 5. In this case, transportation requests are changed, so their pick-up and delivery points are changed simultaneously (changes in pairs), and this guarantees that pick-up point will always be before to its respective delivery point.

```
GIVEN (α, SAmax, T0 e TC) DO
CREATE (a solution S through to distribution heuristic);
APPLY (the programming heuristic in all the routes from S);
S* ← S;                         {Best solution}
IterT ← 0;                      {Iterations number at temperature T}
T ← T0;                         {Current temperature }
WHILE (T > TC) DO
   WHILE (IterT < SAmax) DO
      IterT ← IterT + 1;
      CREATE (any neighbor S' through to one of change moves);
      APPLY (the programming heuristic in all the routes from S');
      Δ ← f(S') – f(S);
      IF (Δ < 0)   S ← S';
         IF (f(S') < f(S*))    S* ← S';   END-IF
      ELSE
         TAKE (x ∈ [0,1]);
         IF (x < e^(-Δ/T))    S ← S';   END-IF
      END-IF
   END-WHILE
   T ← α * T;    IterT ← 0;
END-WHILE
S ← S*;
RETURN (S);
```

Figure 6. Simulated Annealing Algorithm.

Starting from this neighborhood structure, SA was implemented in a way that each neighboring solution is generated for just one of these movements, and its choice is done in a random mode, uniformly distributed, making possible a good diversity among the generated intermediate solutions, and consequently a good exploration of the space of solutions.

The objective function *f(S)* used to evaluate the solutions is described by equations (1), (2) and (3) (see Section 3), and the constraints presented in model proposed in Section 3 ((4) to (18)) are implicitly assisted in distribution and programming heuristics and in change moves described in this section. A pseudo-code of implemented SA is presented in Figure 6.

5. COMPUTATIONAL RESULTS

Several experiments were performed with instances presented by Cordeau & Laporte (2003b) (available in: <http://www.hec.ca/chairedistributique/data/darp />) to evaluate the potential of the presented approach. These instances are references in works of great importance for DARP resolution (see Bergvinsdottir (2004), Bergvinsdottir *et al* (2004) and Cordeau & Laporte (2003b)). 20 instances are available, and they are combined among 24 to 144 transportation requests (48 to 288 points) and 3 to 13 vehicles, and the first instances set (R1a - R10a) are formed by "wide" time windows while the remaining ones (R1b - R10b) are formed by "narrow" time windows.

These instances represent problems with unique depot and homogeneous fleet, and they don't adopt the concept of maximum waiting time. However, the model here proposed (Section 3) adapts easily to them. For that, it is enough to treat the unique depot as origin and destiny depot of all the vehicles, and a very big value as maximum waiting time (that will eliminate the possibility to occur any violation).

The parameters used by SA in all the experiments were T_0 = 20000, α = 0.975, *Tc* = 0.01 and *SAmax* = 1000. The choice of the weights values (vectors ω and β) applied in objective function ((1), (2) and (3)) was based on an analysis presented in Bergvinsdottir *et al* (2004). However, in this work, the weights for violations in essential requirements was "heavier", trying to avoid to get invalid solutions for the problem. The used weights were: ω = [8,0,1,3,1] and β = [1500,1500,1500,1500,1500].

Table 1. Summary of performed tests.

Instance	Vehicles number	Customers number	*f(S)* average	Best *f(S)*	Deviation (%)
R1a	3	24	3721.57	3677.91	1.19
R2a	5	48	7101.87	7017.34	1.20
R3a	7	72	11982.18	11873.76	0.91
R4a	9	96	13982.52	13725.92	1.87
R5a	11	120	16006.82	15736.66	1.72
R6a	13	144	20607.77	20465.39	0.70
R7a	4	36	5718.46	5610.05	1.93
R8a	6	72	11554.71	11343.19	1.86
R9a	8	108	17345.60	15632.09	10.96
R10a	10	144	23104.61	22430.00	3.01
R1b	3	24	3407.87	3379.74	0.83
R2b	5	48	5925.35	5889.56	0.61
R3b	7	72	11045.26	11006.12	0.36
R4b	9	96	12856.75	12807.87	0.38
R5b	11	120	14874.48	14544.13	2.27
R6b	13	144	18795.79	18518.82	1.50
R7b	4	36	5202.93	5136.37	1.30
R8b	6	72	10791.67	10703.17	0.83
R9b	8	108	15180.91	15013.71	1.11
R10b	10	144	20492.56	19969.15	2.62

For validation of the proposed model and SA application to DARP, 5 tests were performed for each instance. Table 1 presents a summary of the obtained results in these tests (*f(S)* is the value of obtained objective function). In this table, the column *Better f(S)* indicates the value of objective function for the best solution found in the 5 tests (for each instance). The column *f(S) average* presents the arithmetic average of the 5 objective functions found, and the column *Deviation* presents the "deviation" obtained among these tests (see equation (22)). In all these tests, the number of used vehicles was the same to the available, and all the transportation requests were assisted, and the most important, all the obtained solutions were valid (all the essential constraints were satisfied).

$$\text{Deviation} = \left(\frac{f(S)\,\text{average} - \text{Best}\ \ f(S)}{\text{Best}\ \ f(S)} \right) * 100 \tag{22}$$

Table 2. Best solutions obtained by Simulated Annealing.

Instance	Traveled distance	Route duration (min)	Waiting time (min) total	avg.	Ride time (min) total	avg.	CPU time (min)
R1a	252.79	831.3	98.51	2.05	241.93	10.08	1.00
R2a	437.45	1992.34	594.9	6.2	310.17	6.46	1.20
R3a	831.74	2404.67	132.93	0.92	894.08	12.42	1.46
R5a	1085.45	3920.25	434.81	1.81	899.35	7.49	1.79
R9a	1064.23	3258.66	34.42	0.16	1275.06	11.81	2.28
R10a	1392.09	4475.42	203.33	0.71	2204.85	15.31	2.72
R1b	251.85	738.42	6.57	0.14	206.66	8.61	0.92
R2b	436.69	1428.44	31.75	0.33	311.95	6.5	1.30
R5b	1010.09	3654.02	243.94	1.02	855.16	7.13	1.95
R6b	1289.31	4318.33	149.02	0.52	1245.66	8.65	1.94
R7b	375.67	1095.67	0	0	345.1	9.59	1.05
R9b	1041.09	3315.28	114.19	0.53	1085.18	10.05	2.26
R10b	1414.65	4332.69	38.04	0.13	1427.08	9.91	2.77
TOTAL	**10883.10**	**35765.49**	**2082.41**	**14.52**	**11302.23**	**124.01**	**22.64**

The best obtained results (Table 2) are still compared to the obtained by Cordeau & Laporte (2003b), through the Tabu Search metaheuristic (Table 3), and compared to the obtained by Bergvinsdottir *et a*l (2004), through a Genetic Algorithm (Table 4). These references were chosen because they represent the current state-of-the-art of DARP. Just the results obtained for some instances are compared to these references, because Bergvinsdottir *et al* (2004) doesn't present the results for the others.

Table 3. Best solutions obtained by Tabu Search.

Instance	Traveled distance	Route duration (min)	Waiting time (min) total	avg.	Ride time (min) total	avg.	CPU time (min)
R1a	190.02	881.16	211.15	4.40	1094.99	45.62	1.90
R2a	302.08	1985.94	723.87	7.54	1976.73	41.18	8.06
R3a	532.08	2579.35	607.27	4.22	3586.68	49.82	17.18
R5a	636.97	3869.95	832.98	3.47	6156.48	51.30	46.24
R9a	672.44	3155.49	323.05	1.50	5621.77	52.05	50.51
R10a	878.76	4480.10	721.33	2.50	7163.71	49.75	87.53
R1b	164.46	965.06	320.60	6.68	1041.50	43.40	1.93
R2b	296.06	1564.74	308.68	3.22	2393.18	49.86	8.29
R5b	589.74	3595.63	605.89	2.52	6104.72	50.87	54.33
R6b	743.60	4072.47	448.88	1.56	7347.39	51.02	73.70
R7b	248.21	1097.25	129.03	1.79	1761.99	48.94	4.23
R9b	601.96	3249.29	487.33	2.26	5581.02	51.68	51.28
R10b	798.63	4040.99	362.37	1.26	7072.29	49.11	92.41
TOTAL	**6655.01**	**35537.42**	**6082.43**	**42.92**	**56902.45**	**634.60**	**497.59**

The results obtained can be compared to the obtained by Cordeau & Laporte (2003b), and can be noticed that the approach proposed in this work had been presented an increase of 63.53% in total distance traveled by the vehicles, however in routes duration, this approach presents an increase of only 0.64%. In average waiting time of the vehicles, the SA approach was able to get a reduction of 66.17%, and 80.46% in average customers' ride time, and the most interesting, a reduction of 95.45% in processing time necessary to get the solutions.

On results obtained by Bergvinsdottir *et al* (2004), the SA approach proposed in this work was able to reduce 11.71% in routes duration, 48.53% in average waiting time, 75.33% in average customers' ride time, and 95.37% in processing time. In this case, the distances traveled by the vehicles are not presented in Bergvinsdottir *et al* (2004), so it is not made any comparison. The distances presented in Cordeau & Laporte (2003b) and here, are Euclidian's distances among the points, and consequently, they don't present measure unit.

Table 4. Best solutions obtained by Genetic Algorithm.

Instance	Traveled distance	Route duration (min)	Waiting time (min)		Ride time (min)		CPU time (min)
			total	avg.	total	avg.	
R1a	-	1039	260	5.42	310	12.90	5.57
R2a	-	1994	514	5.36	1330	27.72	11.43
R3a	-	2781	301	2.09	2894	40.20	21.58
R5a	-	4274	527	2.20	4837	40.30	58.23
R9a	-	3526	32	0.15	6719	62.21	40.78
R10a	-	5025	246	0.86	8341	57.92	65.98
R1b	-	928	164	3.42	549	22.89	5.46
R2b	-	1710	162	1.69	1300	27.07	11.72
R5b	-	4336	568	2.37	4720	39.33	58.93
R6b	-	5227	513	1.78	6397	44.42	81.23
R7b	-	1316	128	1.78	784	21.76	8.29
R9b	-	3676	177	0.82	5358	49.61	44.66
R10b	-	4678	85	0.29	8119	56.38	66.41
TOTAL	**-**	**40508**	**3678**	**28.21**	**51657**	**502.72**	**488.61**

All the tests were performed in a laptop *Toshiba A10 S127* with *Intel Celeron®* of 2.0GHz processor and 256Mb of RAM memory. The whole implementation was developed in C++ language. The solutions obtained by Cordeau & Laporte (2003b) were performed in an Intel Pentium 4 of 2.0GHz PC, and the solutions obtained by Bergvinsdottir *et al* (2004) were performed in a Celeron® of 2.0GHz PC.

6. CONCLUSIONS

This work presented a new approach to solve the dial-a-ride problem. The proposed model was able to represent the problem in a generalized mode, and it was easily adapted to other models already known.

The clustering and routing were performed by the distribution heuristic and neighborhood structure, while the vehicles' programming was performed by the programming heuristic.

The Simulated Annealing, integrated with the other heuristics presented in Section 4, was able to get, in all the cases, and with low processing time, valid solutions for the problem. Besides, it was shown robust, because as can be noted in the Table 1, the presented deviations were satisfactorily low. The neighborhood structure, through the change moves, had been shown to be appropriate and efficient for space of solutions exploration.The obtained results (see Tables 1 and 2) show that the Simulated Annealing, with the proposed model (Section 3) and the other heuristics described in Section 4, were able to produce solutions of good quality for all the instances in expressively low computational times. These results were compared to two other recent approaches found in literature, and in all of cases, the "quality of service" was significantly better. The customers' inconvenience was significantly reduced, which in practice, on human perspective, it reflects better solutions.

Towards the work of Cordeau & Laporte (2003b), the proposed methodology presented an increment in distance traveled by the vehicles, but the routes duration have been almost same. This is justified because of the fact that in results presented by Cordeau & Laporte (2003b) the vehicles "walk" less but they spend good part of their time "waiting", and thus the customers also. In the results obtained by SA, that wait is smaller, because the vehicles prioritize (through the weights) the service to the customers, and the quality of offered service.

Finally, the results show clearly the potential of the presented approach, where solutions of high quality are obtained, for relatively big problems, in expressively low processing times.

Starting from this work, a great research field is had to be explored, as for instance, the application of this approach to real problems found in Brazilian cities, and to other similar problems.

7. ACKNOWLEDGMENTS

The authors acknowledge FAPESP (process 04/11053-9) and CNPq (process 304598/2003-8) for the partial financial support given to the development of this work.

8. REFERENCES

Baugh jr., J.W., Kakivaya, D.K.R. and Stone, J.R. (1998) *Intractability of the dial-a-ride problem and a multiobjective solution using simulated annealing*. Engineering Optimization, 30(2): 91-124.

Bergvinsdottir, K.B. (2004) *The genetic algorithm for solving the dial-a-ride problem*. Master Thesis of Science in Engineering. Department of Informatics and Mathematical Modelling (IMM), Technical University of Denmark (DTU).

Bergvinsdottir, K.B., Larsen, J. and Jorgensen, R.M. (2004) *Solving the dial-a-ride problem using genetic algorithms*. IMM-Technical report-2004-202004. Technical University of Denmark,.

Calvo, R.W., Luigi, F., Haastrup, P. and Maniezzo, V. (2004) *A distributed geographic information system for the daily car pooling problem*. Computers & Operations Research 31: 2263-2278.

Cordeau, J.F. (2004) *A branch-and-cut algorithm for the dial-a-ride problem*. Technical report 2004-23. Center for Research on Transportation (CRT), Montreal, 2004.

Cordeau, J.F. and Laporte, G. (2003a) *The Dial-a-Ride Problem (DARP): Variants, modeling issues and algorithms*. 4OR - Quarterly Journal of the Belgian, French and Italian Operations Research Societies, 1: 89-101.

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

Hart, S.M. (1996) *The modeling and solution of a class of dial-a-ride problems using simulated annealing*. Control and Cybernetics Special Issue on Simulated Annealing Applied to Combinatorial Optimization, 25(1): 131-157, Polish Academy of Sciences Systems Research Institute.

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

Kirkpatrick, S., Gellat, D.C. and Vecchi, M.P. (1983) *Optimization by simulated annealing*. Science, 220: 671-680.

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

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

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

Xu, H., Chen, Z.L., Rajagopal, S. and Arunapuram, S. (2003) *Solving a practical pickup and delivery problem*. Transportation Science, 37(3): 347-364.

Znamensky, A. and Cunha, C.B. (1999) *Um modelo para o problema de roteirização e programação do transporte de deficientes*. Anais do XIII Congresso de Pesquisa e Ensino em Transportes. Rio de Janeiro: Anpet, v. III. p. 59-62.