A contribution on the modeling and resolution of multi-objective Dial a Ride problem

Issam Zidi, Kamel Zidi, Khaled Mesghouni, Khaled Ghedira

A Contribution to the Modelling and the Resolution of a Multi-objective Dial a Ride Problem

Issam ZIDI [1, 2], Kamel ZIDI [1], Khaled MESGHOUNI [2], Khaled GHEDIRA [1]

[1]SOIE : Stratégies d'Optimisation et Informatique intelligentE./ Institut Supérieur de Gestion de Tunis
41, Rue de la Liberté, Cité Bouchoucha 2000 Le Bardo, Tunis –TUNISIE
(zidi.issam@gmail.com, kamel.zidi@fsgf.rnu.tn, khaled.ghedira@isg.rnu.tn)

[2] LAGIS : Laboratoire d'Automatique, Génie Informatique et Signal / Ecole Centrale de Lillle
Cité Scientifique B.P. 48-59651 Villeneuve d'Ascq
(khaled.mesghouni@ec-lille.fr)

ABSTRACT

The paper describes a system for the solution of Dial a Ride Problem (DRP). Dial a Ride Problem (DRP) is to take over the passenger from a place of departure to a place of arrival. It is characterized by a set of transport demands and a number of vehicles available. The ultimate aim is to offer an alternative to displacement optimized individually and collectively. The DRP is classified as NP-hard problem that's why most research has been concentrated on the use of approximate methods to solve it. Indeed, it is a multi-criteria problem, the proposed solution of which aims to reduce both route duration, and cost in response to a certain quality of service provided. In this work, we offer our contribution to the study and solving the DRP in the application using the simulated annealing algorithm. Tests show competitive results on (Cordeau and Laporte, 2003) benchmark datasets while improving processing times.

Keywords*: *Heuristics, Dial a Ride Problem, Multi-criteria Optimization, Simulated Annealing Algorithm

1. Introduction:

Starting with the indisputable observation regarding the increased travel demand of individuals, the available resources are no longer able to satisfy all users. For example, urban public transport is basically affected by their rigidity (ride scheduling would be the application that adapts to the offer). Individual vehicles help avoid the drawbacks of public transport but they are at the same time non-ecological, and costly. So Individuals seeking ways to a more flexible transport that can meet their needs. Indeed the DRP can meet this expectation. It is considered as a collective-individualized transport activated on demand. A DRP consists in meeting the travel demands on a set of passengers scattered geographically. Each transport demand is modelled by a request containing the information on this last. This information is the number of passengers, points of departure and destination, and the intervals between points of time desired.

In every day practice we find different versions of the Dial a Ride Problem; transportation of people in low density areas, transportation of the handicapped and elderly persons, and parcel pickup and delivery service in urban areas are some of the examples. Indeed the DRP have emerged as an area of intense investigations, due to recent advances in communication and information technologies that now allow information to be obtained and processed in real-time (Dror and Powell, 1993; Gendreau and Potvin, 1998; Powell et al., 1995). In the static version of the DRP, it is customary to collect the requests for transportation the day before the beginning of the service.

The DRP belongs to the generic class of vehicle routing and scheduling problems which have been extensively studied over the past 40 years (see, e.g., Toth and Vigo, 2002). It is subject to many constraints and must meet several needs. These needs and / or goals may be contradictory, such as reducing travel time, cost reduction generated, maximizing the quality of service. It is classified as NP-hard problem (Inge Li, 2006) .The exact methods are not able to solve such a problem in a reasonable time, especially as the problem size is important. In this case, we often use methods that find approximate solutions in reasonable time by applying heuristics and meta-heuristics, such as those based on genetic algorithms, simulated annealing, tabu search etc… (Cordeau and Laporte, 2006, Bervinsdottir, 2004, Cordeau and Laporte , 2003, Baugh et al, 1998). In addition, it is a multi-criteria problem. So we need a multi-objective method to solve the DRP.

In this paper, we present the modelization of the transportation problem in the multi-criteria application. Subsequently we apply the simulated annealing algorithm to solve it. The second part of this communication describes the DRP. The third part is the mathematical formulation of the multi-criteria DRP. A description of the algorithm of simulated annealing and the proposed approach for the resolution of this problem is given in part four. In the fifth section, we detail the numerical results obtained that prove the effectiveness of our approach. Finally we present the conclusion and perspectives of this work.

2. The Dial a Ride Problem (DRP)

The DRP is characterized by a set of transport demands of size *"n"* and a number of vehicles *"m"* to serve them. Each transport demand is modelled by a request containing information on demand. To respond to this demand, we must recover a person from a starting point *"i"* and drop it in *"n+i"*. The departure *"i"* must start in the time window $[a_i, b_i]$. Delivery must be made within the time window $[a_{i+n}, b_{i+n}]$. In fact, the DRP is an extension of the Vehicle Routing Problem (VRP) (Debong and Qijun, 2008) (Boudali and al, 2004). Indeed in the DRP, we have an additional constraint which is the consistency of the order of vehicle passage to serve a request. For example, we obviously cannot pass across a point of arrival of a transport demand before carrying the person making the request. So the aim is to design a set of least cost vehicle routes capable of accommodating all requests, under a set of constraints. The most common constraints relate to vehicle capacity, route duration and maximum ride time, i.e., the time spent by a user in the vehicle. In our case, to execute the service, there is a homogeneous vehicles set with the same load capacity that cannot be exceeded. The passengers are picked and delivered by the same vehicle.

In figure 1, we present a simple example of a DRP composed of five transport demands and a fleet of two vehicles. The circles represent the pickups points and the squares represent the deliveries points. Rectangles represent vehicles and arrows represent a vehicle itinerary. The time windows of pickups and deliveries points are represented by the values in brackets. Figure 1 represents at the same time a solution for DRP.

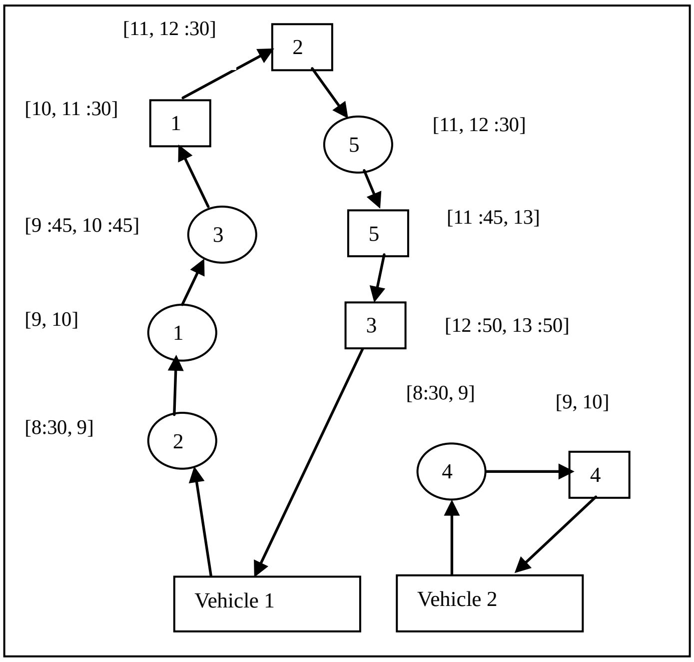

Fig. 1. Schematic representation of a DRP (5 transport requests, 2 vehicles)

2.1 State of the art on the DRP

A DRP is an extension of the PDP (Pickup & Delivery Problem) where the transport of goods is replaced by the transport of persons (Krumke et al, 2006). Several versions of the DRP have been studied over the past 30 years. In the paper (Cordeau and Laporte, 2006), we find a more detailed presentation of the state of the art of this problem. The DRP has been widely studied in literature. In this section, we give a brief literature review on this issue.

There are several variants of the DRP. Indeed, there are DRP with or without time windows and DRP dynamic and static. In the case of dynamic DRP, the problem is usually treated as a succession of static problems (Naba et al, 2004). The majority of research has been focusing on the static DRP but Wilson et al, 1971 have solved the dynamic one.

When the problem size is small, we tend to use exact methods to solve it. In this context we cite the work of Psaraftis who used an exact algorithm of dynamic programming to solve the problem with one vehicle (Psaraftis, 1980). User inconvenience is controlled through a ‘‘maximum position shift’’ constraint limiting the difference between the position of a user in the list of requests and its position in the vehicle route. Only very small instances ($n \leq 10$) can be handled through this algorithm. He studied the case where there are time windows imposed at pickups and delivery points for each request. Still with the exact methods, we find the work of Stefan who solved the DRP using the Branch and Bound method (Stefan and David , 2006). Desrosiers et al, 1991 further improve upon this methodology by performing the insertions in parallel, while Ioachim et al,1995 use a mathematical programming technique to form the clusters. Tests were carried out on instances involving almost 3000 users. Dumas et al,1991 have extended their single-vehicle exact algorithm to the multiple-vehicle case and applied it to instances with $n \leq 55$.

With the increase in travel demands in a DRP, researchers have decided to solve the problem using heuristics and meta-heuristics. These methods enable to reach an acceptable solution to the problem in a reasonable time. In this context, we mention major works such as those of Mauri et al, where the authors have resolved a multi-objective DRP (Mauri et al, 2006). They applied their approach on data derived from the benchmark presented in (Cordeau and Laporte, 2003). Indeed, they have developed a simulated annealing algorithm based on three methods of local search. Cordeau and laporte have applied the tabu search algorithm for solving the problem (Cordeau et al, 2003). Recently Claudio et al, have developed a genetic algorithm for the DRP (Claudio et al, 2009).

For transport problems in the real demand, Garaix et al have developed a method of inserting a transport problem in the application located in a rural area with less density " Pays du Doubs Central, Franche-Comté " (Garaix et al, 2005) . Naba et al, have solved a dynamic DRP using a distributed scheduling algorithm (Naba et al, 2004). This algorithm is applied to a succession of static problems representing the basic problem.

3. Mathematical Formulation Of DRP

The DRP has been modelled mathematically in several research works. It is generally modelled by a multi-objective mathematical program. In this section, we present the mathematical modelling of our DRP. This model is characterized by two main objectives. The first one is economic, and the second is the quality of service rendered to travellers. In this work, we solve a multi-objective DRP using the Simulated Annealing (SA) algorithm in the static context. In the follows part, we present our mathematical formalization of the problem

- **Variables of DRP**

n: Number of transport requests.

$P = \{1, . . . , n\}$: Pickup locations.

$D = \{n + 1, . . . , 2n\}$: Delivery locations

M: set of vehicles depots

$N = D \cup P \cup M$: The set of all nodes in the graph.

Request *i* consist of pickup *i* and delivery $n + i$.

V_i : set of nodes visited by a transport demand i.

V: Set of vehicles

Q_v: Capacity of a vehicle v

$[a_i\ b_i]$: Time window of pickup point of demand

$[a_{i+n}\ b_{i+n}]$: time window of delivery point of demand

q_i : number loaded onto vehicle at node *i*. $q_i = q_{n+i}$.

T_{ijv} : Travel time from *i* to *j* with the vehicle *v*

T_{aiv} : Arrival time for the request *i* with the vehicle v

T_{siv} : Start time of service for the request i with the vehicle v

NSV_i = The number of stations visited by a transport demand *i*.

L_{iv}: The load of vehicle *v* after visiting node *i*

$C_{ijv} = C_{ij} \times C_v$: Cost of travel from *i* to *j* with the vehicle such that *Cv* is the cost of using vehicle *v*

X_{ijv}: Decision variable of the problem,

X_{ijv} = 1 if the vehicle v takes a direct path from i to j, else X_{ijv} = 0

- **The Multi-objective function**

The Multi-objective function: Economic criterion + Service quality criterion

Economic criterion

$$\text{ECO}= \sum_{i\in N}\sum_{j\in N}\sum_{v\in V} X_{ijv} C_{ijv} \tag{1}$$

Service quality criterion:

The Service Quality (SQ) criterion is composed by two major's criteria, the first one is the Ride Time (RT) criterion and the second one is the Number of Stations Visited (NSV) criterion.

.

$$\text{SQ = RT + NSV} \tag{2}$$

RT: Ride Time

NSV: Number of Stations visited

With

$$\text{NSV}= \sum_{i\in D} NSV_i \tag{3}$$

$$\text{RT}= \sum_{i\in D} \sum_{v\in V} (T_{aiv} - T_{siv}) \tag{4}$$

We can rewrite RT (3) using the decision variable X_{ijv}

$$\text{RT}= \sum_{i\in D} RT_i \tag{5}$$

As

$$RT_i = \sum_{i\in V_i}\sum_{j\in V_i}\sum_{v\in V} X_{ijv} T_{ijv} \tag{6}$$

Mathematical Model

$$\text{Minimize } F\,(X_{ijv}) \tag{7}$$

Subject to

$$\sum_{v\in V}\sum_{j\in N} X_{ijv} = 1 \qquad \forall i \in D \tag{8}$$

$$\sum_{j\in D\cup A} X_{ijv} - \sum_{j\in D\cup A} X_{j,n+i,v} = 0 \qquad \forall v \in V, \forall i \in D \tag{9}$$

$$\sum_{i\in N} X_{ijv} - \sum_{i\in N} X_{jiv} = 0 \qquad \forall j \in A \cup D, \forall v \in V \tag{10}$$

$$X_{ijv}(T_{siv} + T_{ijv} - T_{sjv}) \le 0 \qquad \forall v \in V, (i, j) \in N \tag{11}$$

$$a_i \le T_{siv} \le b_i \qquad \forall i \in N, v \in V \tag{12}$$

$$a_{i+n} \le T_{aiv} \le b_{i+n} \qquad \forall i \in N, v \in V \tag{13}$$

$$X_{ijv}(L_{iv} + q_{vj} - L_{jv}) = 0 \qquad \forall v \in V, (i, j) \in N \tag{14}$$

$$q_{iv} \le L_{iv} \le Q_v \qquad \forall i \in D, v \in V \tag{15}$$

$$L_{mv} = 0 \qquad \forall m \in M, v \in V \tag{16}$$

$$X_{ijv} \in \{0;1\} \tag{17}$$

$$F(X_{ijv}) = \alpha1\times \mathrm{ECO}(X_{ijv}) + \alpha2\times \mathrm{RT}(X_{ijv}) + \alpha3\times \mathrm{NSV}(X_{ijv}) \tag{18}$$

The objective function is divided in three parts or objectives (ECO, RT, and NSV). The first part seeks to minimize the economic objective of the problem, while the second one (RT) seeks to minimize the ride time for passenger and the third part seeks to minimize the number of station visited.

In order to handle this multi-criteria objective function each part of the objective function is multiplied by a weight. These weights are denoted α1, α2, α3. The values of the weights are then used to decide the relative weight of each criteria in the overall problem.

To solve the multi-criteria function of the DRP, we apply an aggregative method using a vector of weights α= [α1, α2, α3]. The parameters of this vector are associated to each part of the general objective function.

We used the aggregative method for solving the multi-objective problem, because we model the problem in the general context. Indeed the company will use our algorithm for solving the DR can provide values for the weights associated to the objectives of the problem.

Because the criteria's of the problem are not commensurable, we should normalize it by using a rescaling method. This method consist to divide each criteria of the problem by the optimal value of this last. So the objective function is transformed to this following function:

$$F(X_{ijv}) = k_1\times\alpha1\times \mathrm{ECO}(X_{ijv}) + k_2\alpha2\times \mathrm{RT}(X_{ijv}) + k_3\alpha3\times \mathrm{NSV}(X_{ijv}) \tag{18.1}$$

With $k_1 = 1/\ \mathrm{ECO}(X_{ijv}^{*})$, $k_2 = 1/\ \mathrm{RT}(X_{ijv}^{*})$, $k_3 = 1/\ \mathrm{NSV}(X_{ijv}^{*})$.

Description of constraints

(7): The objective function of the DRP taking into account the quality of service rendered to passengers.
(8): Each customer will be assisted once, for just a vehicle.
(9): A delivery place will always be in the same route that its respective pickup place.
(10): The flow contention (everything that enters is the same to everything that leaves).
(11): Ensures that the arrival time at location j must be later than the sum of departure time from location i and travelling time, $t_{i,j}$ between the locations if that leg is to be part of the route. For example, if vehicle v traverses arc (i, j), where j is a pickup node after service, then its departure time from node *j* is equal to the departure time from the previous node *i* plus travel time $t_{i,j}$.
(12): A vehicle v must satisfy the time window of pickup point i.

(13): A vehicle v must satisfy the time window of Delivery location i + n.
(14): Ensures that the number of passengers passed on a path (i, j) by a vehicle v is conserved.
(15): The number of passengers in the vehicle v after visiting i is higher than that collected in i and less than the maximum capacity of vehicle.
(16): Ensures that the actual loads of the vehicles are set to zero at the depots.
(17): guarantees that decision variables X_{ijv} will be binary
(18): the multi-criteria function of the DRP

(18.1): the multi-criteria function of the DRP with the rescaling constants

The proposed formulation of the DRP is used only to modelise the problem in concern. So this last is not subject to an exact resolution using a plate-form to resolve a mathematical model. We don't adopt this method because the problem is classified from in the NP-hard problem and we cannot solve it with an exact method in a reasonable time .

4. Developed Approach

The hardness of the problem depends on the number of constraints "N". When N is small, traditional mathematical programming approaches can be used to obtain the real optimal solution of DRP; however, when N is large, it is not possible to do that. Therefore, researchers have developed various algorithms that can finish performing within polynomial time to find the problem's initial feasible solution and then apply the meta-heuristic approach to obtain (near) global optimum solution.

To solve a NP-hard problem like the DRP, we do not have polynomial algorithms for their resolutions optimally. Using an approximate method is almost mandatory. In this paper, we applied the simulated annealing algorithm for solving the DRP. The Simulated Annealing (SA) algorithm is a method following the process used in metallurgy. SA algorithm was originated by Kirkpatrick et al,1983 . SA was developed from the so-called "statistical mechanics" idea. Annealing is the process through which slow cooling of metal produces good, low energy state crystallization, whereas fast cooling produces poor crystallization. The optimization procedure of simulated annealing reaching a (near) global minimum mimics the crystallization cooling procedure. SA is classified among the research methods operating locally; it can make changes to the current solution to exit a local optimum. Generally, suddenly reducing high temperature to very low (quenching) cannot obtain this crystalline state. In contrast, the material must be slowly cooled from high temperature (annealing) to obtain crystalline state. During the annealing process, every temperature must be kept long enough time to allow the crystal to have sufficient time to find its minimum energy state. The local search continuously seeks the solution better than the current one during the searching process.

To solve the DRP with the simulated annealing algorithm, a method for generation of an initial solution *S*, a method for generation of neighbouring solutions *S'* (neighbourhood structure), and an objective function *f(S)* to be optimized should be defined. When applying the developed approach to solve the DRP, we need to decide the solution representation of vehicle routes. In our approach, we adopted a matrix presentation of the solution. So, Lines show the vehicles and the columns are the pickup and delivery points. . If *"i"* is the index of rows and *"j"* is the index of columns, *"sol [i] [j]"* represents the passage order of the vehicle "i" by the point *"j"*. The DRP is an extension of the Vehicle Routing Problem (VRP). To solve this problem we adopted the approach of solving routing problems *"two-stage"* in detail (Benjaafar et al, 2006). The general principle of this approach is to divide the problem into two sub-problems. The first is a problem of assigning vehicles to transportation requests. The second is a problem of routing vehicles. In Figure 2 we present the phase of assigning vehicles to the demands for a DRP formed by 5 demands of transport and 2 vehicles. In Figure 3 we present the routing phase as a solution to the problem. Reminding for a transport demand "i", we associate the point of departure number "i" and the arrival point number "i+n".

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	1	1	1	0	1	1	1	1	0	1
2	0	0	0	1	0	0	0	0	1	0

Fig. 2. Assignment of vehicles to transport requests

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	2	1	3	0	6	4	5	8	0	7
2	0	0	0	1	0	0	0	0	2	0

Fig. 3 An example of solution to the DRP

Each vehicle must start from the depot, and then visit the pickups and delivery's points sequentially according to the number in solution representation. According to the constraint of time window, each vehicle arrives at each point in the solution must within point's time window. A vehicle can arrive before the starting of the time window but still needs to wait until the allowable time of pickup or delivery; otherwise, it will violate the time window constraint.

The approach based Simulated Annealing (SA) algorithm developed in this research is composed by 2 major's phases. The first phase is the assignment of transport demands to vehicles. The second phase is the route planning for each vehicle. In the assignment phase we use a classification method based on the approximation of time windows of transportation requests. This phase is used to get an initial solution of problem. The initial solution of the SA algorithm is generated by a distribution heuristic. In the second phase (route planning) we use the local search structure of the SA algorithm to generate the best itinerary for each vehicle. After the generation of a solution for the problem, we apply a programming heuristic (Cordeau and Laporte, 2003) to determinate the starting time of vehicle from the current position. In figure 4, we present the architecture of our developed approach.

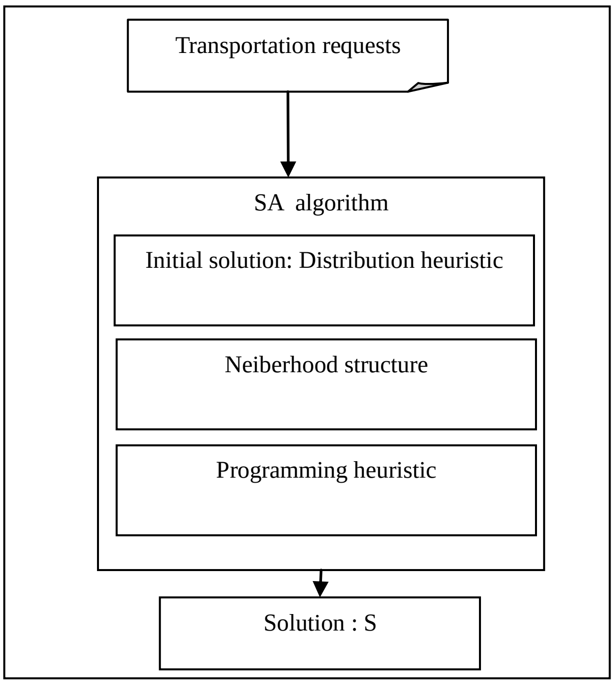

Fig. 4. Developed approach architecture

A pseudo-code of implemented SA is presented in Figure 5.

```
GIVEN ( α , SAmax, T0 , 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);
```

Where x $\in$[0 1]: randomly value used to calculate the probability to move to a neighborhood solution S', 0< α <1 is a cooling rate, Iter-MAX: the number of iteration for each temperature, T0: the initial temperature, TC: the final temperature.

4.1 Initial solution

This study used distribution heuristic to set out the initial solution of SA. Our heuristic based on the method found in the work of Mauri et al, 2006 . That is responsible for routing the vehicles, for forming clusters of places in the routes and determine their attending sequence of these. This heuristic allows the assignment of m vehicles to a set of n transportation requests. This method of finding initial solution based on the principle of random assignment of vehicles to transport requests. After assigning vehicles to requests, they will be ordered randomly in the tour of each vehicle. In this 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 not uniform mode to these routes, and the *n* transportation requests are randomly divided among the *m* vehicles. This heuristic is presented in Figure 6.

```
1.  CREATE (m empty routes for m vehicles)
2.  Create a list L on the set of n queries transport
3.  FOR (each queries transport k, k = 1,2,...,n) DO
4.      Select a random route r  from m routes
5.      Pos1 ← any position of route r;
6.      Pos2 ← any position of route k, but after to
            Pos1;
7.      INSERT (the pick-up point in Pos1);
8.      INSERT (the delivery point in Pos2);
9.      INSERT (the origin depot of vehicle m in
        beginning of route r);

10.     INSERT (the destiny depot of vehicle k at
        the end of route k);
11. END-FOR;
```

Fig. 6. Distribution Heuristic.

In figure 7, we present an example of composition of initial solution. We designate by “i” the pickup point of transportation demand “i,” and “i + n” design the delivery point. Our distribution heuristic may violate same constraints as the vehicle capacity and the respect of time windows on pickups and delivery’s points. Indeed, in our approach we start from an initial solution that does not necessarily satisfy the constraints of the problem. These constraints will be satisfied by the process of finding solutions adjacent to the initial solution using the neighbourhood structure.

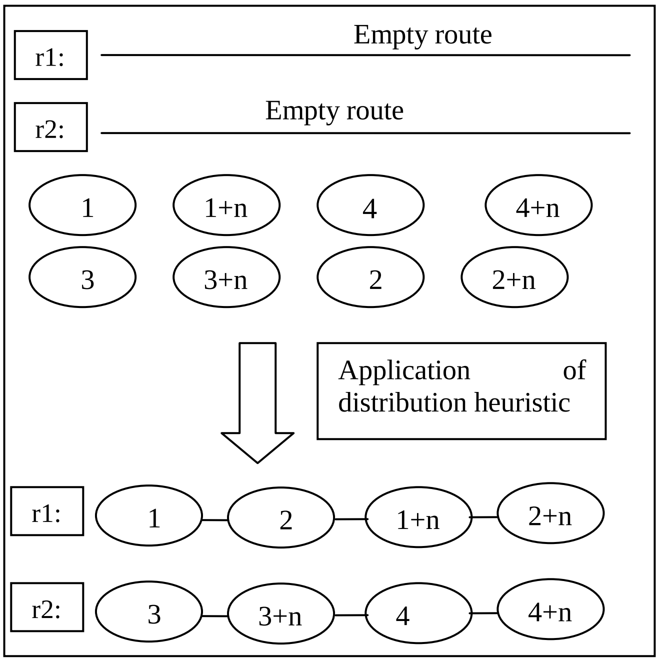

Fig. 7. An example of composition of initial solution

4.2 Neighborhood Structure

To improve the solution of the DRP, we must make changes to the current solution. These changes are made by a neighbourhood structure. In the simulated annealing algorithm can accept change even if they degrade the quality of the solution to escape the local optimum. Our neighbourhood structure is based on the method found in the work (Cordeau and Laporte, 2003). This method is described as follows:

(Let ri and rj two roads that serve the requests i and j)

1. Select two requests i and j;
2. Eject + (remove) the request j from the route rj;
3. Insert the pickup point of the request j in the best position (minimum cost) of the route ri;
4. Insert the delivery point of request j d after the pickup of that in the best position (minimum cost) of the route ri;
5. Eject + (remove) the request i from the route ri;
6. Insert the pickup point of the request i in the best position (minimum cost) of the route rj;
7. Insert the delivery point of request i after the pickup of that in the best position (minimum cost) of the route rj;

In Figure 8, we present an example of changing the solution by the neighbourhood structure used in our approach. Each pickup point for a request "i" is modelled by the number "i" and the delivery point by the number "n + i".

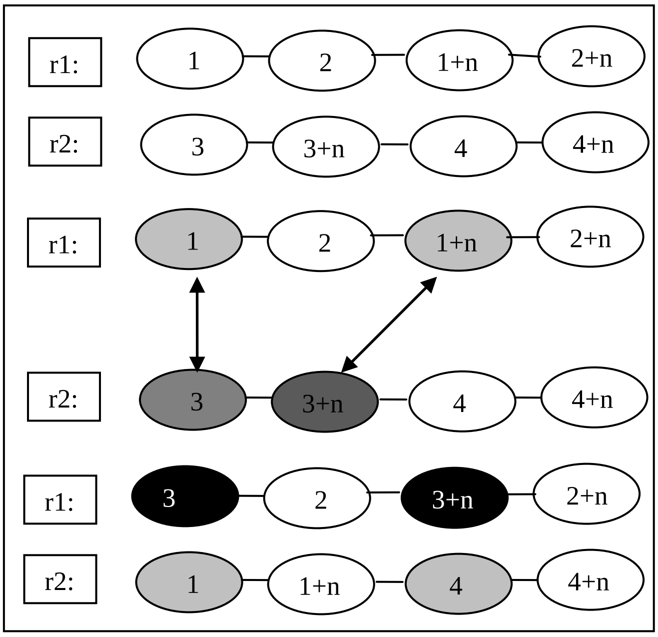

Fig. 8. Neighbourhood Structure

4.3 Programming heuristic

The distribution heuristic and the neighborhood structure are used to program the routes of vehicles, but the programming of these route vehicles should still be made to determine the arrival times in places, the departure times, and so on. Indeed, another heuristic, denominated *programming heuristic* is used. The programming heuristic is adapted of the one presented in (Cordeau and Laporte 2003), and performs the programming trying to reduce the violations in time windows, in routes duration and in ride times. Considering a route $k=(v_0,\dots,v_i,\dots,v_q)$ performed by the vehicle k ($\forall\ k \in K$) where v_0 and v_q both represent the depot. We denote by Ai the arrival time of a vehicle at vertex vi, by $B_i \geq \max\{a_i; A_i\}$ g the beginning of service at vertex v_i, and by Di the departure time from vertex vi. We assume here that waiting at any vertex v_i is allowed before service starts but is not allowed after service has finished. The waiting time of the request i is $W_i =$ Bi-Ai. The ride time associated with request i is computed as Li= B_{i+n}- D_i. With i+n is the delivery point of the request i. So the setting $D_0=a_0$ and $B_i=\max\{ai,A_i\}$ for i=1, . . . , q is optimal in terms of minimizing time window violations because the vehicle leaves the depot as early as possible and the service at each vertex also begins as early as possible. For any route *Vk*, the dela   y is computed as presented in equations (19), (20).

$$\min_{i \leq j \leq q} \left\{ b_j - \left(B_j - \sum_{i \leq p \leq j} W_p \right) \right\} \qquad (19)$$

$$= \min_{i \le j \le q} \left\{ \sum_{i \le p \le j} W_p + (L_j - B_j) \right\} \quad (20)$$

The delay programming heuristic for the delay is the following:

1) Set D0 := B0;
2) Compute Ai, Wi , Bi and Di and for each vertex vi in the route;
3) Compute F0;
4) Set D0 := a_{0+} $\min\left\{ \sum_{0 \le p \le q} W_p , F_0 \right\}$;
5) Update Ai, Wi , Bi and Di for each vertex vi in the route;
6) Compute Li for each request assigned to the route;
7) For every vertex vj that corresponds to the origin of a request j
   a) Compute Fj;
   b) Set Bj := Bj_+ $\min\left\{ \sum_{j \le p \le q} W_p , F_j \right\}$; Dj : Bj +dj;
   c) Update Ai, Wi , Bi and Di, for each vertex vi that comes after vj in the route;
   d) Update the ride time Li for each request i whose destination vertex is after vertex vj
8) Compute changes in violations of vehicle load, route duration, time window and ride time

The steps (1) and (2) of this heuristic minimizes the time window constraints violations. In addition, this heuristic minimizes route duration without increasing time window constraints violations in steps (3) and (6). The step (7) of this heuristic minimizes ride times by delaying the beginning of service at each origin node as much as possible without increasing route duration, time window or ride time constraints violations.

5. Computational Results

In this work, we chose to test our approach to data presented in (Cordeau and Laporte, 2003) (available in: <http://www.hec.ca/chairedistributique/data/darp />). We can compare our results against previously published work. Indeed, in this benchmark, we find 20 instances of DRP. These instances are diversified by the number of transport requests and the number of vehicles. The size of these instances of problems is ranging from 24 to 144 requests. The number of vehicles used to serve the transport demands vary from 3 to 13 vehicles. This benchmark is composed of homogeneous vehicles. These instances represent problems with unique depot they don't adopt the concept of maximum waiting time. However, the model here proposed (Section 3) adapts easily to them. The distance between any two locations "i" and "j" is set to be the Euclidean distance between the coordinates of locations "i" and "j", i, j $\in$ N. The speed of the vehicles is set to 1, so the transportation time "$t_{i,j}$" is equal to the Euclidean distance between "i" and "j".

In Table 1, we give the values of the parameters of the Simulated Annealing algorithm (SA). These values are chosen after several tests applied to problems.

Parameter	Value
T0	3000
A	0.975
TC	0.001
NB-ITER-MAX	1000

Table 1. The parameters of SA algorithm

The best obtained results (Table 2) are still compared to the obtained by (Claudio et al, 2009) and (Cordeau and Laporte, 2003). In the work (Claudio et al, 2009), the authors have applied the genetic algorithm to solve the DRP. In (Cordeau and Laporte, 2003), the tabu search algorithm is applied to solve the problem in concern.

The weights associated with the multi-criteria function "F" already detailed in Section3 are: [0.7, 0.2, 0.1]. These values are chosen for the fact they give good results when comparing it with the other obtained by (Claudio et al, 2009) and (Cordeau and Laporte, 2003).

It is important to mention that in (Claudio et al, 2009), solutions considers a restriction not covered by our approach. These are the time windows as soft constraints.

Table 2 shows the results obtained by our work while Tables 3 and 4 show the results obtained by the previously mentioned researches. Because the compared models do not have the same characteristics, the comparison was done on the basis of time units of two critical factors: On the first hand, the total route duration that is associated with the transport system resources optimization, on the other hand, the total client travel time that is associated with the offered quality of service.

In our tests, the time is in minutes and the travelled distance is in kilometer, rounded to the nearest integer.

Instance	**Best** Route Duration (min)	**Best** Ride Time (min)	CPU time (min)
Pr01	982	590	0,98
Pr02	1840	814	4,70
Pr03	2678	3348	11,07
Pr05	4106	2893	9,61
Pr11	1013	603	0,88
Pr12	1449	774	4,71
Pr15	4062	3099	15,34
Pr16	4632	2311	18,90
Pr17	1439	905	2,32
Pr019	3407	2871	12,03

Table 2. Summary of the results obtained by our approach: SA algorithm

Instance	**Best** Route Duration (min)	**Best** Ride Time (min)	CPU time (min)
Pr01	955,25	524,59	1,36
Pr02	1839.06	838,41	4,08
Pr03	2787.18	1597.95	7,96
Pr05	4068,05	2935,48	18,43
Pr11	902,18	449,91	1,58
Pr12	1503,34	744,93	4,49
Pr15	4057,08	3152,67	22,09
Pr16	4658,64	2348,48	17,48
Pr17	1223,68	612,40	3,13
Pr019	3427,06	2515,53	25,43

Table 3. Summary of the results obtained by Genetic Algorithm (Claudio et al, 2009):

Instance	**Best** Route Duration (min)	**Best** Ride Time (min)	CPU time (min)
Pr01	881	1095	1,9
Pr02	1985	1977	8,06
Pr03	2579	3587	17,18
Pr05	3870	6154	46,24
Pr11	965	1042	1,93
Pr12	1565	2393	8,29
Pr15	3596	6105	54,33
Pr16	4072	7347	73,7
Pr17	1097	1762	4,23
Pr019	3249	5581	51,28

Table 4. Summary of the results obtained by Tabu Search of (Cordeau & Laporte ,2003)

In figure 9 and figure 10, we present the comparison of our approach base on the SA algorithm with the other approach mentioned in this article respectively in terms of route duration and ride time.

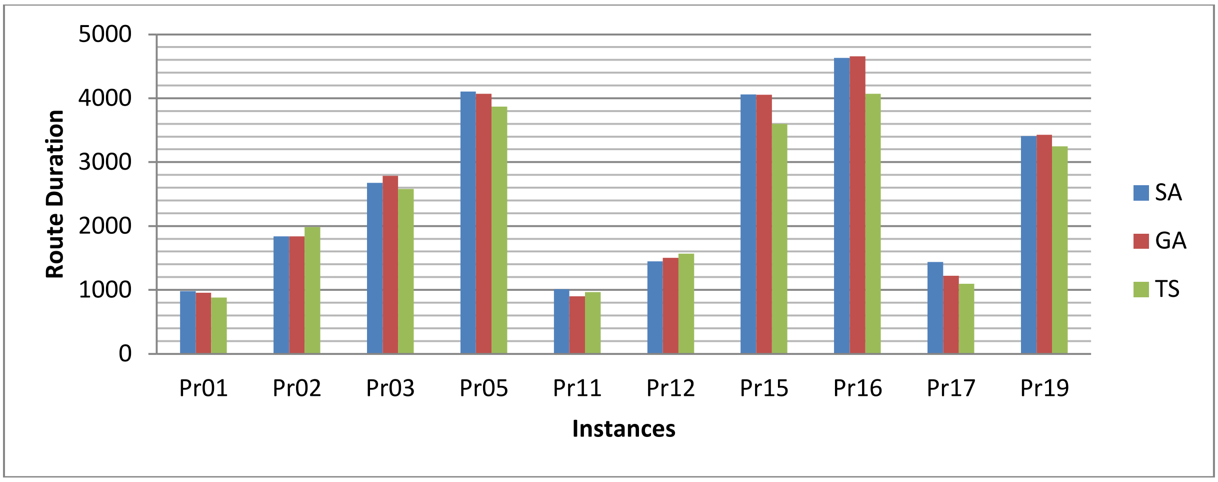

Fig. 9. Route duration comparison

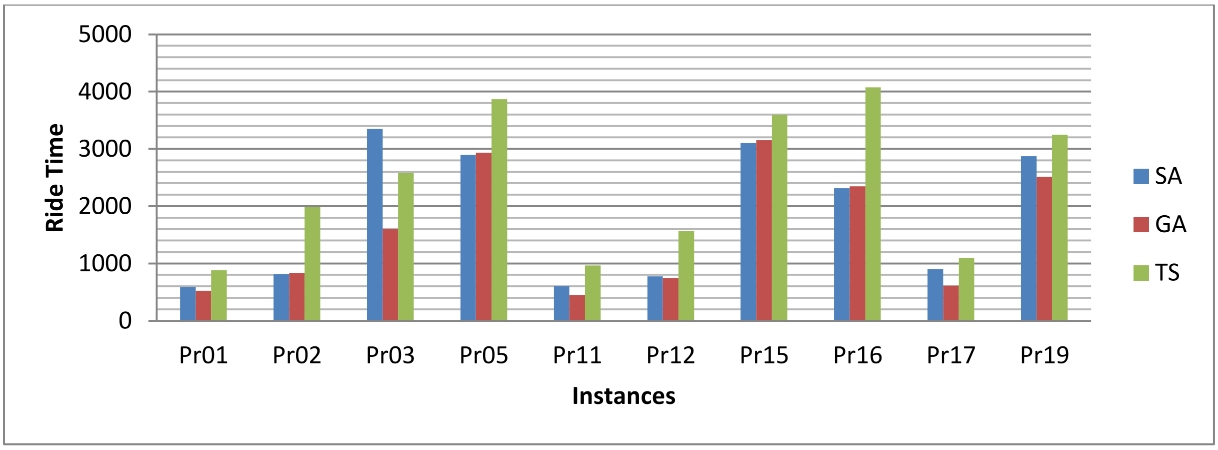

Fig. 10. Ride Time comparison

After presenting our results and results obtained in (Claudio et al, 2009) and (Cordeau and Laporte, 2003), we note that in the some instances, our approach is more efficient than the approach implemented in (Claudio et al, 2009). The values outlined in our summary of results, foresee cases where our approach is better than (Claudio et al, 2009). In fact when we based on the number of individual results for the *Route Duration*, our approach SA shows better results than GA in (6/10) times and worse than TS in (8/10) times. For the *Ride Time*, our approach SA shows better results than TS in (10/10) and equal to GA. when we based on the total time for the *Route Duration*, our approach SA is worse than GA (201 mn) and worse than TS (1763 mn). For the Ride Time, our approach SA is worse than GA (2492 mn) and better than TS (18830 mn).

As exposed in the results section, our SA implementation presents better results than obtained in (Claudio et al, 2009) for the route duration time. This is mainly due to the use of time-windows as hard constraint. But our approach SA shows worse results than TS implementation (Cordeau and Laporte, 2003) for the route duration time. This can be understood by the factor of using an aggregative method with a big weight assigned to the route duration objective to solve the multi-criteria DRP in the TS implementation and the cost of having good average ride times for clients, as there is a trade-off relation among both variables. From other point of view, it can be seen as a search with memory, as in Tabu search. However, in this case the memory is used for making the algorithm to "remember" which portions of sequences are feasible in order to reduce effort instead of remembering the solutions found so far to avoid local optima.

When focusing on the ride time our solution showed equal times (5/10 times) regarding the Genetic Algorithm (GA) of Claudio et al. This can be understood as the use of time-windows as soft constraints in the GA implementation while obtaining better results when compared to the TS solution (Cordeau and Laporte, 2003). This can be understood as the cost of having good average ride times for clients, as there is a trade-off relation among both variables.

It is worth highlighting that the tests were performed in a laptop Dell B14DEE640C with Intel Intel Core 2 Duo of 2.0GHz processor and 2GB of RAM memory. The whole implementation was developed in JAVA language, while the Claudio et al,2009 tests were done with a 2.66 GHz Intel Pentium 4 CPU .The Cordeau and Laporte tests were done with a 2.0 GHz Intel Celeron CPU. Although the hardware configurations are dissimilar, they do not completely justify the time improvement.

6. Conclusion and Perspectives

In this paper, we proposed a mathematical model for the multi-objective DRP. Indeed, we have proposed an approach based on simulated annealing algorithm for solving the DRP. After applying our approach on a benchmark presented in (Cordeau and Laporte, 2003), we have had good results in short computing times. The resulting method was compared to the results given by (Claudio et al, 2009) and (Cordeau and Laporte, 2003) . The comparison focused on the travelled distance and the route duration.

However, improvements can be made about our approach:

- The hybridization of simulated annealing algorithm with other methods of local search (the tabu search and genetic algorithm).
- Resolution of the problem with heterogeneous vehicles to assign each customer to the appropriate vehicle.
- The application of this approach to real problems

References

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).

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. vol 30. pp 91-124.

Bodin, L., Sexton, T., 1986. The multi-vehicle subscriber dial-a-ride problem. TIMS Studies in the Management

Sciences 26, 73–86.

B. Suman and P. Kumar, .A survey of simulated annealing as a tool for single and multiobjective optimization,. Journal of the Operational Research Society, vol. 57, no. 10, pp. 1143.1160, 2006.

Cordeau, J.F. and Laporte, (2006). The Dial-a-Ride Problem: Models and Algorithms. Les Cahiers du GERAD G–2006–78 Copyright c 2006 GERAD

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

Claudio Cubillos, Enrique Urra, Nibaldo Rodríguez,2009. Application of Genetic Algorithms for the DARPTW Problem. Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844 Vol. IV (2009), No. 2, pp. 127-136

Dror, M., Powell, W.B. (Eds.), 1993. Special Issue on Stochastic and Dynamic Models in Transportation. Operations Research 41, 1–235.

Desrosiers, J., Dumas, Y., Soumis, F., Taillefer, S., Villeneuve, D., (1991). An algorithm for mini-clustering in handicapped transport. Cahier du ERAD G-91-22, _EEcole des Hautes _EEtudes Commerciales, Montreal.

Dumas, Y., Desrosiers, J., Soumis, F., (1991). The pickup and deliveryproblem with time windows. European Journal ofOperational Research vol 54, pp 7–22.

Dumas, Y., Desrosiers, J., Soumis, F., 1989. Large scale multi-vehicle dial-a-ride problems. Cahiers du GERAD G-89-30, E´ cole des Hautes E´ tudes Commerciales, Montre´al, Canada.

Gendreau, M., Potvin, J.-Y., 1998. Dynamic vehicle routing and dispatching. In: Crainic, T.G., Laporte, G. (Eds.), Fleet Management and Logistics. Kluwer, Boston, pp. 115–226.

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

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

Michel Nabaa, Besma Zeddini, Pierrick Tranouez, (2004). Approche décentralisée pour résoudre le problème du transport à la demande , MajecSTIC2004

Madsen, O.B.G., Ravn, H.F., 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.

Mauri G.R, Lorena L.A.N, (2006). A Multiobjective Model and Simulated Annealing Approach for a Dial-a-Ride Problem. Workshop dos Cursos de Computação. http://mtc18.sid.inpe.br/col/sid.inpe.br/ePrint@80/2006/10.09.16.10/doc/v1.pdf

Powell, W.B., Jaillet, P., Odoni, A., 1995. Stochastic and dynamic networks and routing. In: Ball, M.O., Magnanti, T.L., Monma, C.L., Nemhauser, G.L. (Eds.), Network Routing. North-Holland, Amsterdam, pp. 141–295.

P. Toth, D. Vigo (2002).The Vehicle Routing Problem with Backhauls.In P. Toth, D. Vigo (Eds.).The Vehicle Routing Problem, SIAM Monographs on Discrete Mathematics and Applications,Philadelphia, 195-224.

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

Savelsbergh, M.W.P., Sol, M., 1995. The general pickup and delivery problem. Transportation Science 29, 17–29.

Stefan Ropke and David Pisinger, 2006, An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows TRANSPORTATION SCIENCE 40: 455-472.

S. O. Krumke, W. E. de Paepe, D. Poensgen, M. Lipmann, A. Marchetti- Spaccamela and L. Stougie,(2006). On minimizing the maximum flow time in the online dial-a-ride problem, volume 3879 Springer, Berlin/Heidelberg

T. Garaix, D. Josselin, D. Feillet, C. Artigues, and E. Castex. Transport à la demande points à points en zone peu dense. proposition d'une méthode d'optimisation de tournées. In SAGEO'2005, 2005.

Wang Dedong and Chen Qijun, (2008). Stochastic Vehicle Routing Problems and their Solution Algorithm. Proceedings of the IEEE International Conference on Automation and Logistics Qingdao, China

Wilson, N.H.M., Sussman, J., Wong, H., Higonnet, B., (1971). Scheduling Algorithms for dial-a-ride systems. Technical Report USL TR-70-13, Urban Systems Laboratory, Massachusetts Institute of Technology, Cambridge, MA.