A NEW HEURISTIC APPROACH FOR DEMAND RESPONSIVE TRANSPORTATION SYSTEMS

Rui Jorge Gomes
Faculdade de Engenharia da Universidade do Porto
Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
rjgomes@fe.up.pt

Jorge Pinho de Sousa
Faculdade de Engenharia da Universidade do Porto / INESC Porto
Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
jsousa@fe.up.pt

Teresa Galvão Dias
Faculdade de Engenharia da Universidade do Porto
Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
tgalvao@fe.up.pt

ABSTRACT

Providing quality public transportation is extremely expensive when demand is low, variable and unpredictable. Demand Responsive Transportation systems address this problem with routes and frequencies that may vary according to the observed demand. In this context, we aim at planning a set of services for transportation requests, between origins and destinations specified by users, using a fleet of homogeneous vehicles. Users may also specify a time window for departure and the desired arrival time. The goal is not only to minimize operating costs but also to maximize the quality of the service, expressed by indicators such as the average passenger waiting time and on-board time. To obtain an approximation of the Pareto solution set for this problem, we have designed a heuristic approach involving the construction of a feasible route through a greedy randomized procedure, followed by a local search phase. Preliminary results on randomly generated instances look very promising.

KEYWORDS. Combinatorial Optimization. Multiple-Objectives. Heuristics. Logistics and Transportation.

1. Introduction

Providing quality public transportation is extremely expensive in scenarios where demand is low, variable and unpredictable, such as disperse rural areas or some periods of the day in urban areas (e.g. at night). Buses circulating with very low occupancy rates mean higher costs for the service providers, often leading to low frequencies and, as a consequence, low perceived quality and degradation of the image of public transportation. Demand Responsive Transportation (DRT) services try to address this problem by providing a kind of hybrid approach between a taxi and a bus, with routes and frequencies that may vary according to the actual observed demand. Metaphorically speaking, one could see a DRT as a "horizontal lifter". The advantages of such a service in terms of social cohesion, mobility, traffic, or environment, are fairly obvious. However, in terms of financial sustainability and quality level, the design of this type of services may be rather difficult.

The problems of designing and operating DRT services are closely related to the Vehicle Routing Problem (VRP), and in particular to the Dial-A-Ride models. Given the complexity of these problems, optimal solutions can take an enormous amount of time to be found, ruling out their usefulness in the context at hand. Besides, in a multiple criteria decision context the "optimal" solution is in general meaningless because it is impossible to satisfy all (usually contradictory) objectives simultaneously (Branke et al. 2008). So we are interested in finding a set of efficient solutions hopefully close to the Pareto front.

The Vehicle Routing Problem is a NP-Hard combinatorial optimization problem, dating back to the 50's (Dantzig and Ramser 1959) , that lies at the intersection of two well known and studied problems (Machado et al. 2002): the Travelling Salesman Problem (TSP) and the Bin Packing Problem (BPP). In the TSP, one is interested in finding a minimum Hamiltonian circuit. In the BPP, the problem is to pack a set of items of different size and/or weight in a container not exceeding its maximum capacity. In VRPs, given a limited fleet of vehicles, a depot as starting and ending point and the known demands of geographically dispersed clients, the objective is to find the set of routes with minimum cost satisfying all the demand (Fisher et al. 1995).

Dynamic Vehicle Routing for Demand Responsive Transportation extend the "classical" VRP in a number of ways, being, at least, as much complex as the later (Cordeau et al. 2007a). It is clear that in the DRT context, vehicles have a limited capacity (leading to a variant of the so-called Capacitated VRP), demands should be served in a certain time window (VRP with Time Windows), each stop along the route can be both a pickup and delivery point (Pickup-and-Delivery VRP) and there is still the uncertainty and variability associated with the number of stops along the route – thus making the problem dynamic. But, DRTs drift away from "classical" VRP in the sense that, instead of the vehicles leaving the depot loaded to serve the demand along the route, in DRTs they leave the depot empty and have several points of pickup and delivery along the route, before returning to the depot.

There is a more suitable class of problems for modelling the DRT, known as the Dial-A-Ride Problem (DARP) (Cordeau and Laporte 2007). In the DARP model, one tries to define vehicle routes and schedules for a set of transportation requests, between origins and destinations specified by the users. This transportation requests are performed by an homogeneous fleet of vehicles starting from a depot, providing a shared service in the sense that several users may be in a vehicle at the same time (Cordeau 2006). The biggest difference between the DARP and the VRP (and namely the Pickup-and-Delivery VRP variant) is, what we might call, the human dimension of the problem: in the DARP one is interested not only in minimizing the operating costs or the distance travelled by the vehicles but also (and this is sometimes more important) in maximizing the quality of the service, based on indicators such as the average passenger waiting time or the on-board (ride) passenger time (Paquette et al. 2010). In the DARP there are usually outbound and inbound trips (Cordeau et al. 2007b), in what we could call pendulum movements, but in our DRT problem there are just outbound trips – i.e. users specify pickup and delivery locations, but no return trip. Also, in most literature DARP instances, the set of pickup points is disjoint from the set of delivery points (in some cases there is only one delivery point) which, clearly, is not the case in the addressed problem as we shall see.

In this document we present a Dynamic Vehicle Routing approach for Demand Responsive Transportation - DVRDRT. Besides being a multi-objective DARP problem, the DRT application is also dynamic, requiring the (re-)design of solutions in real-time, and this is strongly dependent on the dynamic degree of the problem at hand. According to the concept of effective degree of dynamism, for problems with time time-windows as defined in (Larsen 2000), the studied DRT services are strongly dynamic.

2. Problem description

In the Dynamic Vehicle Routing model for DRT we assume that passengers specify origins and destinations from a set of pre-defined possible route points, a pickup time window and a desired arrival time for their transportation needs, and that they are to be served by a fleet of vehicles of equal capacity (number of seats). Each possible route point, with the exception of the depot, can be a pickup-only point, a delivery-only point, or both. At a given route pickup location, different passengers entering the vehicle can have different destinations and different time windows. Several users can be simultaneously transported in one vehicle, like a mini-bus. The vehicles start and end their trips at a single depot and transportation requests can be received at any time, from any origin. Since different users have different transportation needs, each point (stop) along the route can have multiple (possibly disjoint) time-windows (both pickup and delivery), which in association with the real-time arrival of new requests may require several visits to a given stop at different periods. This is a major difference from all know variants of the VRP and DARP problems – and quite a fundamental one, thus requiring innovative approaches. Summing up, the main DVRDRT characteristics are:

- multiple vehicles with equal capacity;
- single depot where vehicle routes start and finish;
- simultaneous pickup and delivery;
- users specify transportation requests from anywhere to any where (many-to-many), at any time (dynamic);
- users specify pickup and delivery time-windows;
- multiple (possibly overlapping) time-windows at each stop;
- pickup time-windows must be respected (hard constraint);
- delivery time-windows can be violated at a penalty cost (soft constraint).

For combinatorial optimization problems as this, one is often “just” interested in feasible solutions that can be obtained in useful time. Given the complexity of the problem, optimal solutions can take an enormous amount of time to be found, ruling out their usefulness in the context at hand. Besides, in a multiple criteria decision analysis the concept of an “optimal” solution is in general complex to define because it is impossible to satisfy all (usually contradictory) objectives at the same time (Branke et al. 2008). So we are interested in finding a set of efficient solutions hopefully close to the Pareto front. The goal is not only to minimize the operating costs incurred to satisfy the maximum possible number of requests but also to maximize the quality of the service, expressed by indicators such as the average passenger waiting time and the on-board (riding) time. To obtain an approximation of the Pareto solution set, we have designed a heuristic approach by constructing a feasible route through a greedy random approach, followed by a local search phase.

The figure below shows a hypothetical initial state for a static DVRDRT problem (i.e., with all information know beforehand). In a dynamic environment, during the time period in which vehicles perform the pre-computed routes, new transportation requests would stochastically arrive in real time from any point.

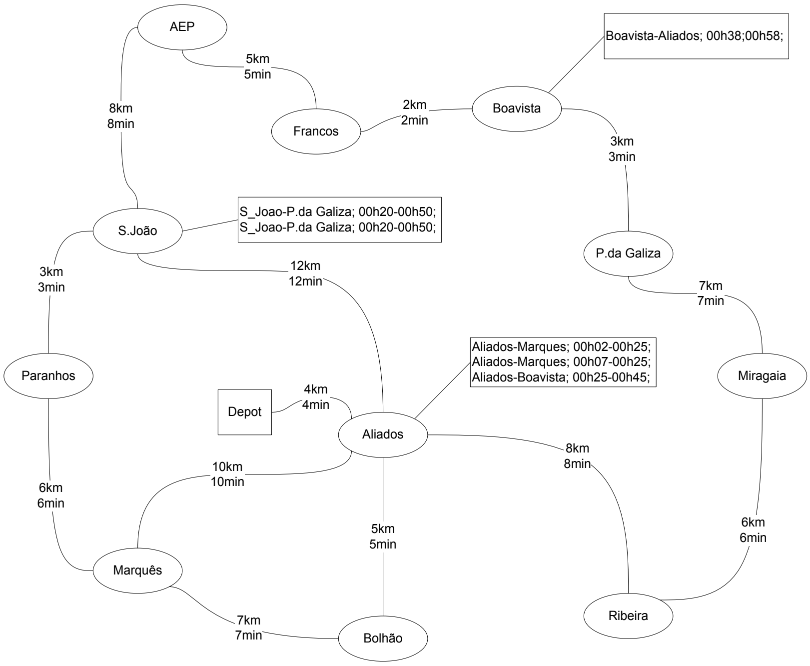

Figure 1 Hypothetical initial state for a static DVRDRT problem

Suppose that in the hypothetical situation depicted above, the service operator has set the start of the service at midnight (00h00) and the time-windows have 3 minutes. There are a set of stops represented as ellipses, the road network represented as arcs with distance and travel time connecting the stops, one depot (square) and the transportation requests represented as tags at their origins. Each transportation request is formed by origin, destination, pickup time and delivery time. Around each value of pickup and delivery time there is a (customizable) time window. In the above picture, there are, for instance, two transportation requests with origin in "S. João" and destination "P. da Galiza", with the same pickup and delivery time-windows, [00h20m-00h23m] and [00h50m-00h53m], respectively. At "Aliados", for instance, there are three transportation requests, two of them with the same destination and similar pickup times and one with a completely different destination and pickup time (i.e., multiple destinations and multiple time windows). Notice also that "Boavista" is simultaneously a pickup and delivery stop for different transportation requests also with different time windows.

3. Problem modelling

In (Cordeau 2006) the static DARP is formally modelled using a 3-index formulation. We will use a similar formulation for the DVRDRT problem also, noting that in this case there is no distinction between pickup and delivery nodes – i.e., every node can be both pickup and delivery - and there is a single depot. The presented formulation is intended to be flexible, easily allowing new constraints and variants of the problem.

3.1 Parameters and indices

Let $P = \{1,2,.., p\}$ be the set of transportation requests. As each passenger corresponds to one and only one request, we can use interchangeably the terms passenger and request. So, for each passenger $p \in P$ there is a transportation request from origin p^+ to destination p^-. So $P^+ = \bigcup_{p \in P} p^+$ is the set of all origins and, analogously, $P^- = \bigcup_{p \in P} p^-$ is the set of all destinations. The dynamic vehicle routing problem for demand responsive transportation can therefore be defined on a loop less asymmetric multigraph $G=(V,A)$. $V=\{0,1,..,n\}$ is the node set representing the stops, being 0 (zero) the vehicles' depot, so $V=\{0\}\cup P^{+}\cup P^{-}$. $A=\{(i,j): i\neq j, i,j\in V\}$ is the set of arcs connecting the vertices. Associated to each arc $(i,j)\in A$ there is a traversal cost c_{ij} and a trip time d_{ij}. The set of all possible trips corresponds to the non-reflexive closure of $V\setminus\{0\}$. To each vertex one can assign a load index m_i equal to the difference between the number of requests having this vertex as origin and the number of requests having this vertex as destination, i.e., $m_i=|P_{in_i}|-|P_{out_i}|$ with $P_{in_i}=\left\{\bigcup_{p\in P} p^{+} : p^{+}=i, i\in V\setminus\{0\}\right\}, P_{in_i}\subseteq P^{+}$ and, in a similar way, for the delivery, $P_{out_i}=\left\{\bigcup_{p\in P} p^{-} : p^{-}=i, i\in V\setminus\{0\}\right\}, P_{out_i}\subseteq P^{-}$. There are no passengers entering or leaving at the depot $P_{in_0}=P_{out_0}=\phi$.

As referred, each passenger (request) p has a pickup time-window $\left[e_{i_p}, l_{i_p}\right]$ at the origin i and a delivery time-window $\left[e_{i_p}, l_{j_p}\right]$ at the destination j, $i\neq j, i,j\in V$.

Let $K=\{1,2,..,k\}$ be the set of vehicles, all with the same capacity (number of seats) Q. Q_i^k is the load of the vehicle k after leaving node i. $U=\{1,2,..,u\}$ is the set of the requests that have not been satisfied.

3.2 Decision variables

$x_{ij}^k\in\{0,1\}$ is 1 if vehicle $k\in K$ travels from node i to node and j 0 otherwise

$y_p^k\in\{0,1\}$ is 1 is passenger $p\in P$ enters vehicle $k\in K$ and 0 otherwise

3.3 Auxiliary variables

For each vehicle $k\in K$, the continuous variable $t_{i_p}^k$ is the time instant of the beginning of the service for passenger p at node i. The service duration is considered to be null. If passenger p is not served by vehicle k then this value has no meaning.

3.4 Objective functions

Minimize cost $$\min \sum_{k\in K}\sum_{i\in V}\sum_{j\in V} c_{ij} x_{ij}^k \quad (1)$$

Minimize waiting time $$\min \sum_{k\in K}\sum_{p\in P}\left(t_{i_p}^k - e_{i_p}\right) y_p^k \quad (2)$$

Minimize on-board ride time $$\min \sum_{k\in K}\sum_{p\in P}\left(t_{j_p}^k - t_{i_p}^k\right) y_p^k \quad (3)$$

Maximize number of serviced requests $$\max \sum_{k\in K}\sum_{p\in P} y_p^k \quad (4)$$

3.5 Constraints

Several constraints have to be considered. In particular, we have to impose that every vehicle that enters a node leaves that node, routes start and end at the depot, and every transportation request can only be served by a single vehicle. Moreover, neither the fleet size nor the vehicle capacity can be exceeded. Every passenger must be picked up within his pickup time window (hard constraint) and, preferably, also delivered within the limits of the specified delivery time window - delivery time-windows can be violated at a penalty cost (soft constraint). In every transportation request the delivery service must, of course, be later than the pickup service (precedence constraints).

3.6 Solutions

A feasible solution for the DVRDRT is a set $S = \{R_1,.., R_m\}$ of tuples R formed from the vertex set V that respect the problem's constraints. S is not a partition of V, as in (most) VRP problems, because there is no guarantee that $R_1,.., R_m$ are both collectively exhaustive and mutually exclusive with respect to the set V being partitioned. R_i is a tuple since it can contain an object (vertex) more than once (i.e., it is not a set), the objects appear in a certain order (i.e., it is not a multiset) and, obviously, they have finite size. Also, although the visiting order is a key feature of the problem, each R_i is not a permutation of the vertices of the problem's graph as in the VRP because each vertex may have to be visited more than once. It is not a permutation with repetitions in the strict sense because the repetitions of a node cannot be consecutives – i.e., the vehicle must move from a node to another. So, a feasible R_i must contain at least two nodes, besides the depot, corresponding to the origin and destination of at least one transportation request $\min R = < 0, p^+, p^-, 0 >$. As the starting and finishing point of a route is the depot, the minimum size of a feasible route must be at least 4, and the theoretical maximum size of a feasible route is given by $|P^+| + |P^-|$ (in this case, $|S| = 1$).

Each route $R_i = < 0, v_1,.., v_m, 0 >$ has a cost function $C(R_i) = (\sum_{i=1}^{m} c_{i,i+1} + W)$, with W being the fixed cost of the vehicle assigned to the route. Similarly, each request not satisfied has a cost $C(U_i)$. So, finally, the total cost of a solution S is $F(S) = (\sum_{i=1}^{m} C(R_i) + \sum_{j=1}^{u} C(U_j))$.

4. Heuristic approach

A generic solution strategy has been developed for efficiently solving the problem. For testing and validating this approach, a simplified version (to be extended in the near future) of the problem has been designed based on the following assumptions:

- a single depot from which a set of uncapacited vehicles starts and finishes the routes;
- each passenger specifies a pickup time window and a corresponding delivery time window;
- time windows have a fixed, common duration (e.g., 15 minutes);
- transportation requests must be issued before the start of the service in order to have an *a priori* and static route planning. Transportation requests that arrive after the vehicle starts the service are not accepted for the ongoing route and are postponed for

a later service, if feasible.

4.1 A greedy constructive algorithm

The problem objectives are classified into two perspectives: a vehicle's perspective and a passengers' perspective. From the vehicle's perspective we have the minimization of total route cost and the maximization of the serviced requests, and on the passengers' perspective (reflecting the quality of service) we have the minimization of the sum of passenger waiting times and the sum of passenger ride times.

A Node Ranking Function (NRF) has been defined to determine, at each iteration, the next node to be inserted into the route (under construction), taking into account the two aforementioned perspectives. In terms of the vehicle's perspective, the major factors for determining the next node to be selected are the distance to all other nodes from the current position and the number of passengers on those nodes. From the passengers' perspective, the major factors to be considered are the number of passengers on the bus having as destination a given node, and the time windows on the remaining nodes. For each of these factors a weight α is assigned, to account for the different perspectives of the decision maker in a multi-criteria context. Let α_d be the weight of the distance factor, α_p the weight of the number of passengers' factor, α_v the weight of the delivery time window factor and, finally, α_t the weight of pickup time window factor. The NRF can then be defined as:

$$\forall i \in V \setminus \{0\}, NRF[i] = (\alpha_d \times CRL[i] + \alpha_p \times NRL[i]) + (\alpha_v \times DRL[i] + \alpha_t \times TRL[i])$$

with the first operator representing the vehicle's perspective and the second operator representing the passengers' perspective. The $CRL[i]$ (Cost Rank List) is the node's travel cost normalized value, the $NRL[i]$ (Number of passengers Rank List) is the node's load index m_i normalized value, $TRL[i]$ (Delivery time rank list) is node's earliest delivery time normalized value and $DRL[i]$ (Time window rank list) is the node's associated earliest pickup time normalized value.

The normalization of the values is obtained setting $(Z_i^k)_N = \dfrac{Z_i^k - Z_{\min}^k}{Z_{\max}^k - Z_{\min}^k}$, for maximization and $(Z_i^k)_N = \dfrac{Z_{\max}^k - Z_i^k}{Z_{\max}^k - Z_{\min}^k}$, for minimization. The NRF algorithm is as follows:

Step 1: initialize $S = \{ \}$

While $P \neq \phi \wedge K \neq \phi$

- **Step 2:** initialize $R = \{ \}$
- **Step 3:** start at the depot $R = \{0\}$
- **Step 4:** for all feasible nodes:
  - **Step 4.1:** compute *Cost Rank List* (CRL) - sort all nodes in increasing distance from the current position, and normalize the values obtained, such that the closest node is assigned with the highest value, and so on;
  - **Step 4.2:** compute *Number of Passengers Rank List* (NRL) – sort all nodes in decreasing order by load index m_i and normalize.
  - **Step 4.3:** compute *Delivery time-window Rank List* (DRL) – sort the nodes that have delivery requests ($P_{outi} > 0$) by increasing earliest delivery time defined by the time window plus the trip time from the current position to every remaining node and normalize.
  - **Step 4.4:** compute *Time-window Rank List* (TRL) - sort the nodes that have pickup requests ($P_{ini} > 0$) by increasing earliest pickup time defined by the time window plus the trip time from

the current position to every remaining node and normalize.

Step 5: set NRF for each node, according to

$$\forall i \in V \setminus \{0\}, NRF[i] = (\alpha_d \times CRL[i] + \alpha_p \times NRL[i]) + (\alpha_v \times DRL[i] + \alpha_t \times TRL[i])$$

Step 6: select the node with highest NRF that does not violate the constraints (feasible node) and add it to the route - $R = R \cup \max(NRF[i])$;

Step 7: update requests data, possibly removing requests already satisfied and moving the unfeasible ones to the list U;

Step 8: **if** $P = \phi$ **then** close route R adding the depot (0) at the end and add route to solution set S, $S = S \cup R$;

Step 9: **if** $U \neq \phi$ **then** do $P = U$ and go to Step3; **else** go to next;

end-while

return solution S.

The next figure shows the route solution set obtained using the NRF algorithm in the hypothetical initial state for the static DVRDRT problem instance shown in Figure 1, with the weight parameters equal to: $\alpha_d = 0{,}20, \alpha_p = 0{,}15, \alpha_v = 0{,}55, \alpha_t = 0{,}10$,

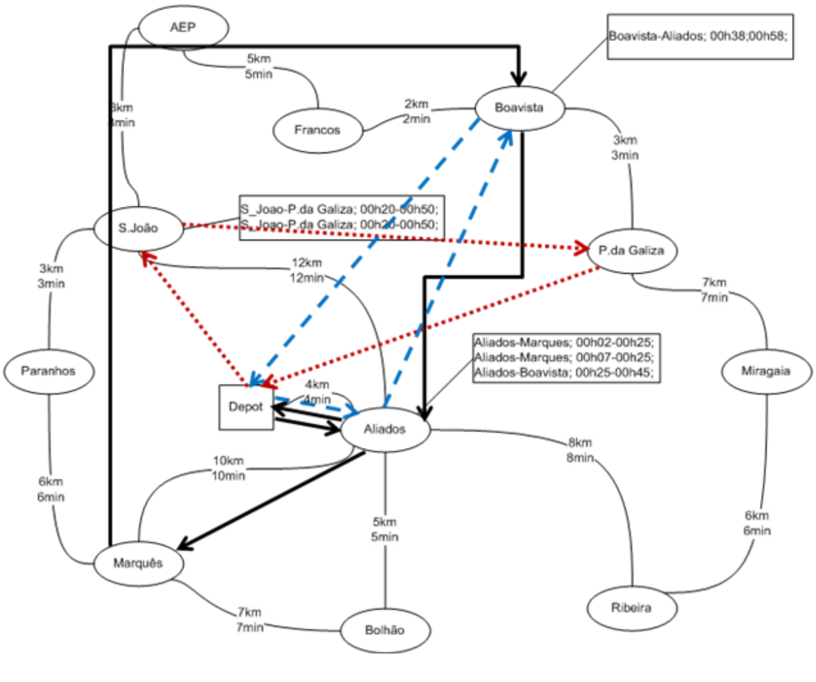

Figure 2 NRF algorithm solution

The NRF algorithm solution for the hypothetical initial state for a static DVRDRT problem is a set of three routes:

- Depot /Aliados / Marquês / Boavista /Aliados / Depot
- Depot / S.João / P.da Galiza / Depot
- Depot /Aliados / Boavista / Depot

4.2 Randomized enhanced heuristic

In the quest to obtain better results, hopefully closer to the Pareto front, the next step was to lessen the myopic-greedy nature of the NRF algorithm, embedding the NRF in a randomized choice procedure for the next node to be inserted on the route being constructed. The result is a GRASP-like metaheuristic. As in other metaheuristics, the Greedy Randomized Adaptive Search Procedure (GRASP) (Feo and Resende 1989), at each iteration constructs an initial solution (construction phase) and them performs local search procedures to improve that initial solution (local search phase). The difference to other metaheuristic lies at the construction of the initial solution in a greedy, randomized and adaptative manner (and that's where its name comes from). The focus of this metaheuristics is in the attempt to build the best possible initial solution and not so much in the local search phase. The construction strategy is to evaluate the elements to be inserted in the solution at each iteration according to some criteria – recall that the DVRDRT is a multi-criteria problem. These criteria adapts to the already built solution, such that the evaluation of the elements changes during the construction of the solution. In the process, there is a random choice between the best elements according the defined criteria at each iteration. The evaluation of each element according to the criteria is made the NRF function. We have yet to address the address the local search phase.

In the construction phase, a feasible solution (set of routes) is built by applying the NRF algorithm, adding to the each initially empty route one element at a time. Each NRF algorithm iteration constructs a candidate list (CL) of the elements to be inserted in the current route. From this CL a number of its best elements are selected to form a restricted candidate list (RCL) - $RCL \subseteq CL$. The size of the RCL is defined by a parameter $\alpha \in [0,1]$ that sets either the numbers of elements or a threshold between the value of best element of the CL and the value of the last element to be included in the RCL. This last approach was considered the best for the DVRDRT problem because, being a very constrained problem and due to the adaptative nature of each iteration of the algorithm, the size of the CL varies and, sometimes, has very few elements. By setting the α parameter to zero, only the best element from the CL is select, in a "pure greedy" manner, while setting the parameter to 1 it will be completely random. We have implemented a memory scheme to learn the appropriate value for the α parameter that controls how random and greedy the construction process following the Reactive GRASP (Resende and Ribeiro 2003) implementation. The algorithm reacts to solutions produced using different values for the α parameter and tries to adjust it (at every 200^{th} algorithm iteration) to give the "best" balance between greediness and randomness. The next step in the algorithm is to randomly select one element from the RCL in insert it in the route being constructed. When a route cannot satisfy any more transportation request, the route is finished. If there are any unsatisfied feasible requests left and other vehicles available, a new route for another vehicle is started and built in the same manner. The process is repeated until there are no more feasible transportation requests left to satisfy or no more available vehicles (it is useful to recall at this stage that, if one has enough vehicles at hand, every feasible request can be satisfied assigning a "individual" vehicle to it). The final solution is the resulting set of routes and its cost is calculated. The found solution could then used in the local search phase. It is a multi-start metaheuristic, so each iteration returns a solution $S = \{R_1,..,R_m\}$ with its cost. Only the best overall solution is kept as the final result.

The randomized, enhanced algorithm is as follows:

```
Parameters:MAX_iterations
while(num_iterations<MAX_iterações)
        choose α_k parameter with probability p(α_k), k = 1,..,m
        initialize S = { }
        //construction phase
        while P ≠ φ ∧ K ≠ φ
                initialize R = { }
                start at the depot R = {0}
                while P ≠ φ
                        Build Candidate List (CL) from current node using NRF
                        Build Restricted Candidate List (RCL) using current α_k
```

Randomly select next node from RCL - $RCL[i]$
Check feasibility of next node according to problem' constraints
if (feasible) **then**:
update requests data, possibly removing requests already satisfied and moving the unfeasible ones to the list U
add it to the route - $R = R \cup \max(NRF[i])$
"move" to this position
else:
add this node to a "unfeasible nodes for current route" list
end-while
close route: **if** $P = \phi$ **then** close route R adding the depot (0) at the end and add route to solution set S , $S = S \cup R$;
if $U \neq \phi$ **then** do $P = U$
end-while
compute solution cost $F(S) = (\sum_{i=1}^{m} C(R_i) + \sum_{j=1}^{u} C(U_j))$
//update α parameter probabilities at each 200th iteration
if mod(num_iterações,200)==0 **then** $p(\alpha_k) = q_k \Big/ \sum_{j=1}^{m} q_j , k = 1,..,m$
update best solution found so far S^* : **if** $F(S) < F(S^*)$ **then** $S^* = S$
end-while
return best solution S^*

The figure below shows the route solution set obtained using the Randomized-enhanced NRF algorithm in the hypothetical initial state for the static DVRDRT problem shown in Figure 1, with the weight parameters equal to: $\alpha_d = 0{,}20, \alpha_p = 0{,}15, \alpha_v = 0{,}55, \alpha_t = 0{,}10$,

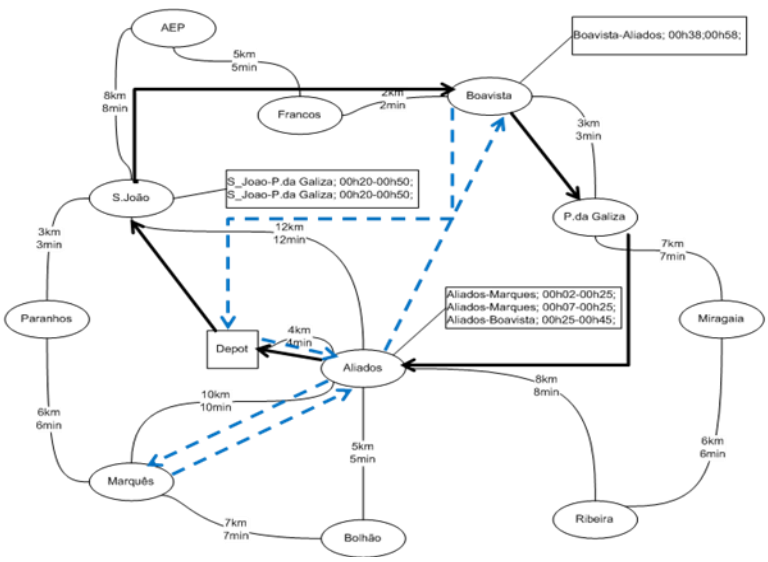

Figure 3 Randomized enhanced NRF algorithm solution

The solution produced by the randomized algorithm for the example is a set of two routes with significant smaller cost than the solution obtained with the "pure greedy" NRF algorithm (see Figure2 for comparison):

- Depot / S.João / Boavista / P.da Galiza / Aliados / Depot
- Depot / Aliados / Marquês / Aliados / Boavista / Depot

4.3 Computational results

Being a "new" problem, there are no "off-the-shelf" benchmark data bases to test the algorithm for the DVRDRT and to compare it with other published approaches. To the best of our knowledge, the most similar instances in the literature are the ones for the Capacitated VRP with Time Windows (e.g. Solomon 1987), the Capacitated VRP with Pick-up and Deliveries and Time Windows (e.g. Haibing and Lim 2001) and the Dial-A-Ride-Problem (DARP) (e.g. Gilbert and Cordeau). But, even if the two problems are similar, at least two adaptations need to be made: one on the DVRDRT program to accept a different input format, and a second adaptation in the benchmark database itself to convert it to a DVRDRT instance. Our decision was to use randomly generated instances.

Computational tests were done using an Intel Core Duo running at 1,67GHz, 2GB RAM memory, and the adjustment of the α parameter that controls greediness/randomness level at every 100th algorithm iteration. Preliminary computational results on these instances look very promising, both in terms of cost savings and in terms of computational efficiency. These results seem to highlight that the major factor affecting the algorithm running time is the number of passengers. The following picture, obtained using 50 stops and 1000 algorithm iterations shows the effect of increasing the number of passengers (requests):

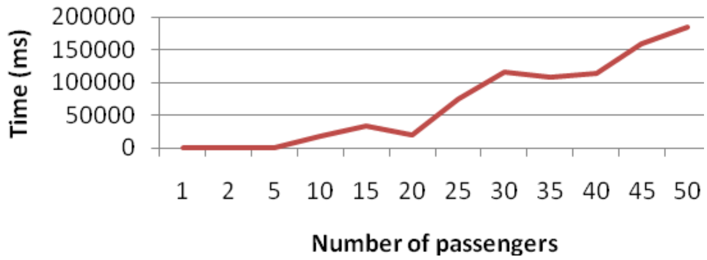

Figure 4 Number of passengers effect on the randomized enhanced NRF algorithm

Moreover, if the number of passengers is fixed, adding possible stops does not increase the algorithm running time. Another observation is the linear increase in running time with the number of iterations, the running time for each iteration being constant – this is in line with literature results for GRASP-based algorithms. The next picture captures this observation for a problem with 50 stops and 20 transportation requests, gradually adding 1000 iterations to the algorithm.

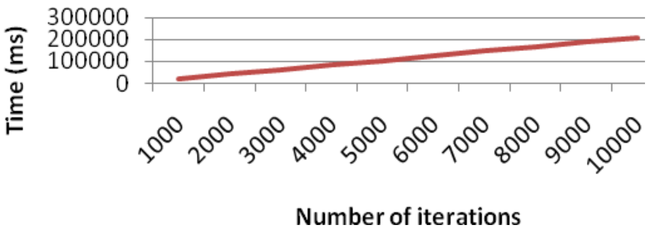

Figure 4 Number of iterations effect on the randomized enhanced NRF algorithm

5. Preliminary conclusions

Providing quality public transportation is extremely expensive when demand is low, variable and unpredictable. DRT services try to address this problem by providing a kind of hybrid approach between a taxi and a bus, with routes and frequencies that may vary according to the actual observed demand. The advantages of such a service in terms of social cohesion, mobility, traffic, or environment, are fairly obvious. However, in terms of financial sustainability and quality level, the design of this type of services may be rather difficult.

The problems of designing and operating DRT services are closely related to the Vehicle Routing Problem (VRP), and in particular the Dial-A-Ride models. Given the complexity of these problems, optimal solutions can take an enormous amount of time to be found, ruling out their usefulness in the context at hand. Besides, in a multiple criteria decision analysis the “optimal” solution is in general meaningless because it is impossible to satisfy all (usually contradictory) objectives simultaneously. So we are interested in finding a set of efficient solutions hopefully close to the Pareto front. The approach proposed in this work seems to be a powerful and flexible tool to model quite different DRT services. The constructive, heuristic algorithm developed here allows for different weights for each factor to be set at the beginning of the process or, more interestingly, at each iteration (thus “changing” the neighbourhood structure). Solutions are sensitive to both the weighs and the rank scale values used.

Preliminary computational results on randomly generated instances look very promising, both in terms of cost savings and in terms of computational efficiency.

References

Branke, J., Kalyanmoy D., Miettinen, K., Słowiński, .R, (eds.) (2008), *Multiobjective Optimization: Interactive and Evolutionary Approaches* (Springer-Verlag).

Cordeau, J. F. and Laporte, G. (2007), 'The dial-a-ride problem: models and algorithms', *Annals of Operations Research*, 153 (1), 29-46.

Cordeau, J.F., Laporte, G., Savelsbergh, M., Vigo, D. and Barnhart, C. (2007a), 'Chapter 6 Vehicle Routing', *Handbooks in Operations Research and Management Science* (Volume 14: Elsevier), 367-428.

--- (2007b), 'Chapter 7 Transportation on Demand', *Handbooks in Operations Research and Management Science* (Volume 14: Elsevier), 429-66.

Cordeau, J. F. (2006), 'A Branch-and-Cut Algorithm for the Dial-a-Ride Problem', *Oper. Res.*, 54 (3), 573-86.

Dantzig, G. B. and Ramser, J. H. (1959), 'The Truck Dispatching Problem', *Management Science*, 6 (1), 80-91.

Feo, Thomas A. and Resende, M. (1989), 'A probabilistic heuristic for a computationally difficult set covering problem', *Operations Research Letters*, 8 (2), 67-71.

Fisher, M., M.O. Ball, T. L. Magnanti C. L. Monma, and Nemhauser, G. L. (1995), 'Chapter 1 Vehicle routing', *Handbooks in Operations Research and Management Science* (Volume 8: Elsevier), 1-33.

Laporte, G. and Cordeau, J. F. (2010), 'VRP data', <http://neumann.hec.ca/chairedistributique/data/>, accessed 04.

Haibing, L. and Lim, A. (2001), 'A Metaheuristic for the Pickup and Delivery Problem with Time Windows', in Lim Andrew (ed.), (0), 160-60.

Larsen, A. (2000), 'The Dynamic Vehicle Routing Problem', PhD (Technical University of Denmark).

Machado, P., Tavares, J., Pereira, F. and Costa, E. (2002), 'Vehicle Routing Problem: Doing It The Evolutionary Way', in W. B. Langdon, et al. (eds.), *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference* (Morgan Kaufmann Publishers).

Paquette, J., Cordeau, J.F., and Laporte, G. (2010), 'Quality of service in dial-a-ride operations', *Computers & Industrial Engineering*, In Press, Corrected Proof.

Resende, M. and Ribeiro, C. (2003), 'Greedy Randomized Adaptive Search Procedures', in Fred Glover and Gary A. Kochenberger (ed.), *Handbook of Metaheuristics* (International Series in Operations Research & Management Science, 57: Springer New York), 219-49.

Solomon,. M. (1987), 'Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints', *Operations Research*, 35 (2), 254-65.