CONTRIBUTIONS FOR THE DESIGN AND OPERATION OF DEMAND RESPONSIVE TRANSPORTATION SYSTEMS

Rui Jorge Gomes
Jorge Pinho de Sousa
Teresa Galvão Dias
Faculdade de Engenharia da Universidade do Porto

RESUMO
Fornecer transporte público de qualidade em cenários de baixa procura é extremamente caro. Sistemas de transportes flexíveis procuram endereçar este problema através de rotas e horários que podem variar consoante a procura observada. Neste contexto, é apresentada uma abordagem flexível inovadora para planejar diferentes tipos de serviços para pedidos de transporte entre dois pontos especificados pelo utilizador. O objectivo é não só minimizar os custos de operação do serviço, mas também maximizar a sua qualidade. A simulação pode ser uma ferramenta importante para estudar como diferentes formas de operar um serviço afectam a sua eficiência. Para obter um conjunto de soluções próximas da fronteira de Pareto foi desenhada uma abordagem heurística de construção sequencial, gulosa e aleatorizada de uma rota admissível, seguida de uma fase de melhoramentos locais. Os resultados obtidos para instâncias geradas aleatoriamente são bastante promissores. Este algoritmo está a ser integrado num Sistema de Apoio à Decisão.

ABSTRACT
Providing quality public transportation is extremely expensive in low, variable and unpredictable demand scenarios. Demand Responsive Transportation systems try to address this problem with routes and frequencies that may vary according to observed demand. This work presents an innovative approach for these systems. We aim at planning a set of services for transportation requests, between origins and destinations specified by users. The goal is not only to minimize operating costs but also to maximize the service quality. Simulation can be used to study how different ways of operating the service affect its performance and efficiency. To obtain an approximation of the Pareto solution set for this problem, we have designed a parallel heuristic that constructs a feasible route through a reactive greedy random approach, followed by a local improvement phase. Preliminary results on randomly generated instances look very promising. This algorithm is being embedded in a Decision Support System.

1. INTRODUCTION

Transportation systems are a key factor for economic sustainability and social welfare, but the economic efficiency of public road transportation strongly relies on solid demand levels and well-established mobility patterns. Providing quality public transportation is extremely expensive when demand is low, variable and unpredictable, as it is the case of disperse rural areas or some periods of the day in urban areas (e.g. during the night). Buses circulating with very low occupancy rates mean high 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 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. Due to this added flexibility, the service provided by the operators becomes more efficient, with routes planned shortly before their start, with better occupancy rates and vehicles with characteristics better suited to users' mobility requirements. The DRT systems can also operate in a complementary way to other transportation systems (Nelson 2004) in the sense that they can be used to feed traditional regular systems in strategic points of the network, thus improving overall public transportation quality while also increasing the number of passengers. 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, and in particular to the Dial-A-Ride models.

The Vehicle Routing Problem (VRP) 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, Tavares et al. 2002): the Travelling Salesman Problem (TSP) and the Bin Packing Problem (BPP). 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, M.O. Ball et al. 1995). Vehicle Routing Problems for Demand Responsive Transportation extend the "classical" VRP in a number of ways, being, at least, as much complex as the later (Cordeau, Laporte et al. 2007). It is clear that in the DRT context, vehicles have a limited capacity, demands should be served in a certain time window, each stop along the route can be both a pickup and delivery point, there is uncertainty and variability associated with both the number of stops along the route and the link travel times. There is a more suitable class of problems for modeling 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 a 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 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, Cordeau et al.). Dial-a-Ride services can operate in a static or dynamic mode. In the static mode, all requests are known before-hand, whereas in the dynamic mode transportation requests are gradually revealed along the service operating time, with routes and schedules having to be adjusted to meet the demand (Psaraftis 1995). In practice, however, "pure" dynamic services are not common since some requests are usually known *a priori*. The importance of dynamic vehicle routing is increasing: logistic distribution scenarios where the information is revealed during the operations are more common and also real-time data processing is easier and less costly.

When designing a DRT service, it is not only important to be able to solve the underlying model in an efficient way, but also understand how different ways of operating the service affect customers and operators. Such effects are often studied by simulation. How the optimization and simulation phases relate to each other can be seen from two perspectives. The first is to find an optimal solution to a specific case, and then simulate what effects this solution has on the performance, customer behavior, and other key performance indicators. The second perspective is to find a good overall design by the use of simulation, and then use optimization to find the best solution to a specific instance of the given design.

In this work we present an innovative approach for DRT services – here referred as Dynamic Vehicle Routing for Demand Responsive Transportation (DVRDRT). Besides being a multi-objective problem, this DRT application is also strongly dynamic (Larsen 2000), requiring the (re-) design of solutions in real-time. Given the complexity of these problems (Lenstra and Kan 1981), 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 conflicting) objectives simultaneously (Branke, Deb et al. 2008). In this context we aim at developing a general modeling framework for planning and managing transportation services of this type. A further subsidiary aim of this research is to develop a Decision Support System (DSS) for providing a set of efficient solutions hopefully close to the Pareto front, by using efficient, customizable multi-objective algorithmic approaches to deal with the combinatorial nature of the problem and with the multiple perspectives of its different stakeholders. The approach was to tackle the design problem by the use of simulation, and then use the developed DSS for providing a set of efficient solutions taking into account the perspectives of the stakeholders. The main goal is not only to minimize the operating costs incurred to satisfy all requests but also to maximize the quality of the service, expressed by indicators such as the average passenger waiting time and the on-board time (Paquette, Cordeau et al.).

2. PROBLEM DESCRIPTION

The main elements for the definition and operation of a DRT system are, as pointed in (Nelson 2004), the routes (their flexibility and the density of links between origins and destinations), the schedules (fixed or flexible), quality factors, the mechanism for collecting passengers, fleet management and the interoperation with other systems. The type of service is partially defined by the flexibility of the route characterized by a sequence of stops. In terms of flexibility, the routes can be classified in three major categories: semi-fixed routes, flexible routes and virtual flexible routes. In flexible and semi-fixed routes the service departs and ends at fixed points at prescribed times. In a virtual flexible route there is no fixed end or intermediate stop points and no fixed schedules. In terms of stops, one can distinguish: terminals, fixed stops (like conventional bus stops), predefined stops (meeting points with or without a predefined passing time) and non-predefined stops (door to door service). In a conventional public transport system these elements (sequence of stops and schedule) are defined in advance. In the service of a taxi, only the origin and destination are set in advance by request of the passenger. For DRT systems a wide spectrum of different combinations of these concepts is possible: starting from a predefined route and timetable to a service with stops and passing times determined during operation. In this context we aim at developing a general modeling framework for planning and managing transportation services of this type.

The Dynamic Vehicle Routing for DRT is similar to the DARP presented in (Madsen, Ravn et al. 1995), assuming 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) – in (Madsen, Ravn et al. 1995) passengers only specify one (and only one) of the possible time windows. 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. 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 for pickup and for 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 known variants of the VRP and the DARP problems – and quite a fundamental one, thus requiring to be handled in a new way. Summing up, the main characteristics for the DVRDRT as considered in this work are:

- multiple vehicles with equal capacity;
- single depot where vehicle routes start and finish;
- simultaneous pickups and deliveries;
  - users specify transportation requests from anywhere to anywhere (many-to-many), at any time (dynamic)
- 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).

3. SERVICE DESIGN

We will consider some stochastic data associated to flexible transportation systems and, in particular, the Dynamic Vehicle Routing for DRT systems will be characterized by data on:

- space: the spatial distribution of the transportation requests, i.e., one wants to know the probabilities of the geographical locations of origins and destinations;
- time: birth time of the transportation requests, i.e., one wants to know when requests are made by the passengers, i.e., the arrival rate/process of requests to the system;
- travel: expected travel time between two points in the network.

Of these three stochastic aspects, only the expected travel time seems to have an impact in our algorithm for operating DRT services. In our work, transportation requests are either known beforehand and/or arrive in a Poisson manner in real time. In the latter case, the algorithm is re-run each time a new request arrives, so the spatial distribution of the new requests and their birth time do not influence the gap between the algorithm outcome and the "real" service operation. This does not happen for the travel time: there is, very often, considerable uncertainty about how long it will take to travel between any two points in a city - due to traffic fluctuations (this is especially true under peak traffic conditions), accidents, changes in weather conditions, road works, unpredictable events, and so on. So if one accounts for the expected travel time instead of a deterministic travel time, the planned routes can be quite different. The stochastic information seems to be more related with service design.

When designing a DRT service, it is not only important to be able to solve the model in an efficient way, but also understand how different ways of operating the service affect customers and operators. Simulation can be used as a tool to study how different ways of operating the service affect its performance and efficiency in a given scenario. Our approach aims at finding a good overall design by the use of simulation, and then use the developed DSS for providing a set of efficient solutions hopefully close to the Pareto front, by using efficient, customizable multi-objective algorithmic approaches to deal with the combinatorial nature of the problem and with the multiple perspectives of the different stakeholders.

4. SERVICE OPERATION

Dial-a-Ride services can operate in a static or dynamic mode. In the static mode, all requests are known before-hand, whereas in the dynamic mode transportation requests are gradually revealed along the service operation and routes have to be adjusted to meet the demand (Psaraftis 1995). As already mentioned, the importance of dynamic vehicle routing is increasing because logistic distribution scenarios where the information becomes available during the operations are more and more common and, on the other hand, real-time data processing is easier and less costly.

A traditional approach for the dynamic problem is to solve static scenarios (Psaraftis 1995) when a new request arrives – each new request creates a new static scenario. However, the routing algorithm must be fast enough to (re)calculate a solution in the case where requests arrive in a quick sequence. When a new request arrives at a given time instant, a route planning system must deal with the request and, possibly, calculate new routes. Route sections already traversed until the arrival of the new request are, obviously, unchangeable. Thus the problem is to re-optimize the remaining part of the initial solution after the insertion of the new request(s), taking into account that all the previous feasible requests are already in the on-going routes and can be in one of three states: "not yet picked up", "picked up but not delivered", "picked up and delivered". On problems with time windows constraints, the insertion of a new request in real-time is more complex: sometimes this new request has to be refused because is not possible to include it in any routes or have another vehicle available to start a new route. As already said, besides being a multi-objective problem, the DRT approach in this work is also strongly dynamic, requiring the (re-)design of solutions in real-time. These problems are NP-hard and therefore optimal solutions cannot be reached in useful time. We have therefore designed a parallel heuristic approach that constructs a feasible route through a reactive greedy random approach, followed by a local improvement phase. In order to "involve" the experts in the planning process, a prototype of a Decision Support System embedding the heuristic approach is being developed.

5. HEURISTIC APPROACH

The Vehicle Routing Problem is a NP-Hard combinatorial optimization problem. Exact algorithms can only solve very limited instances of the problem with extremely variable computation times. Moreover, population based algorithms usually do not exhibit a performance level suitable for the real-time solution generation needs of the problem at hand. We have therefore designed a greedy randomized sequential constructive heuristic to obtain an initial route solution set, followed by an improvement phase. Our main effort was devoted to build the highest quality possible solutions in the construction phase. One appealing characteristic of our heuristic implementation is that it was implemented in parallel, with only a single global variable required to store the best solution found over all processors.

5.1. Construction phase: a greedy constructive algorithm

Each feasible transportation request is composed by an origin, a destination and pickup and delivery times. Having a set of requests, the algorithm tries to find a set of trip sequences (routes) considering the objectives and respecting all problem constraints. The problem objectives are classified into two perspectives: a vehicle's perspective and a passengers' perspective. On the vehicle's perspective we have the minimization of total route cost and the maximization of the serviced requests, and on the passengers' perspective 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 find, at each iteration, the next "best" 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 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 the pickup time window factor. Let also W be the set of all nodes defining the problem and NW the subset of nodes not yet in the solution routes. The NRF is defined as:

$$\forall i \in NW, NRF[i] = (\alpha_d \times CRL[i] + \alpha_p \times NRL[i]) + (\alpha_v \times DRL[i] + \alpha_t \times TRL[i]) \qquad (1)$$

CRL (Cost Rank List) is an ordered list of the normalized travel costs to each node - $CRL[i]$is the cost from the present node to node $i \in NW$. NRL (Number of passengers Rank List) is the ordered list of the normalized number of passengers at each node - $NRL[i]$ is the value at node i. DRL (Delivery Time Rank List) is the ordered list of normalized delivery lower time limit at each node, so $DRL[i]$ is the value associated to the node i. TRL (Time Rank List) is the list of normalized pickup lower time limit at each node, so $TRL[i]$is the value at node i.

Step 1:inicialize $S = \{\ \}$
while $P \neq \phi \wedge K \neq \phi$
    Step 2:inicialize $R = \{\ \}$
    Step 3: start at the depot $R = \{0\}$
    Step 4: compute the NRF rank of all feasible nodes:
        Step 4.1:build the *Cost Rank List* (CRL) – sort all nodes by increasing distance from the current one, and normalize the values obtained, such the closest node is assigned with the highest value and so on;
        Step 4.2:build the *Number of Passengers Rank List* (NRL) – sort all nodes by decreasing order of the load index m_i and normalize.
        Step 4.3:build the *Delivery time-window Rank List* (DRL) – sort all nodes with delivery requests ($P_{out\ i} > 0$) by increasing order of the closest delivery time associated to the node plus the trip time to that node and, finally, normalize so that the "earliest" gets the highest score and so on.
        Step 4.4:build the *Time-window Rank List* (TRL) – sort all nodes with pickup requests ($P_{ini} > 0$)by increasing order of the closest pickup time associated to the node plus the trip time to that node and, finally, normalize so that the "earliest" gets the highest score and so on.
    Step 5:compute NRF for each node, such that
$$\forall i \in NW, NRF[i] = (\alpha_d \times CRL[i] + \alpha_p \times NRL[i]) + (\alpha_v \times DRL[i] + \alpha_t \times TRL[i])$$
    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, eventually removing the ones already satisfied (picked up and delivered), i.e., $|P| = |P| - 1$, and moving the unfeasible ones to a temporary list U
    Step 8:if $P = \phi$ **then** add the depot node (0) to the end of the route R and add this route to the solution set S , $S = S \cup R$;
    Step 9:if $U \neq \phi$ **then** let $P = U$ and **goto** Step3; **else goto** next step;
end-while
return solution S

Figure 1: Pseudo-code for the Node-Ranking Function algorithm

The node with the highest NRF is added to the route under construction at the end of each iteration and the process is repeated.

5.2. Improvement phase

Solutions produced in the greedy randomized construction phase are not necessarily optimal. The improvement phase tries to improve the constructed solution S by exploring "close" solutions, i.e., solutions in some "neighborhood" of the current solution. But the DVRDRT is highly constrained and the combination of simultaneous pickup and delivery with the possibility of having multiple pickups and / or deliveries at each stop, increases significantly the complexity of possible local search procedures. The definition of neighborhood structures is non-trivial and their implementation is computationally very complex (Kindervater and Savelsbergh 1997).

For the improvement phase, we use a combination of three improvement mechanisms: the forward slack time, a "nearby-stops" analysis, and a simple 2-exchange procedure. After calculating the forward slack time ("dead" times available through the route), the "nearby-stop" analysis takes each route in the solution set and "reproduces" its sequence of stops one-by-one trying to find in-between stops that appear later in that route and can be served in the meantime. Suppose, for instance, that a vehicle has been assigned the route [A,C,B,D], the "nearby-stop" analysis detects that B is in the physical path from A to C and when the vehicle leaves A heading to C checks if it is possible to satisfy the request at B on route to C without destroying the time-windows and precedence constraints at any stop. The last improvement is a simple 2-exchange procedure based on the k-interchange procedure by (Psaraftis 1983).

5.3. A GRASP type metaheuristic

In order to produce better solutions, hopefully closest to the Pareto front, the presented NRF algorithm was embedded in a GRASP type (Feo and Resende 1989) metaheuristic. One appealing characteristic of a GRASP implementation that we explored, mainly because our need of real time solutions, is that it can be trivially implemented in parallel, with each GRASP iteration being performed in parallel with only a single global variable required to store the best solution found over all processors. We have implemented a parallel Reactive-GRASP (Resende and Ribeiro 2003) given its suitability for the problem at hand, its relative computational simplicity and good results in terms of performance and solutions quality, as reported in the literature for other problems.

The construction strategy is to evaluate the elements to be inserted in the solution at each iteration according to some criteria defined by the NRF function – the problem objectives. As new nodes are added to the solution routes, these criteria adapts to the already built solution, such that the evaluation of the elements changes during the construction of the solution. Instead of always choosing the "best" node (absolute highest NRF value), in the process there is a random choice between the best elements. This initial solution is then used in local improvements in a first-best procedure. This two phases are repeated a specified number of iterations in parallel.
Next we present the high level pseudo-code of the Parallel Reactive-GRASP for the Dynamic Vehicle Routing for Demand Responsive Transport (DVRDRT) problem, where at every 200th iteration the probabilities of the (α) parameter that controls "reactiveness" are updated.

```
Parameters: GRASP_max_iterations
while (num_iterations<GRASP_max_iterations)
        choose α_k parameter with probability p(α_k), k = 1,..,m
        initialize S = { }
        Construction phase:
                Calculate S using NRF and α_k for the RCL
        If mod(num_iterations,200)==0 then p(α_k) = q_k / Σ_{j=1}^{m} q_j , k = 1,..,m
        Calculate the solution cost F(S) = (Σ_{i=1}^{m} C(R_i) + Σ_{j=1}^{u} C(U_j))
        Improvement phase:
                using S do:
                        forward slack time
                        nearby stops analysis
                        simple 2-exchange procedure
                until F(S') < F(S) or elapsed_time>allowed_running_time
                if F(S') < F(S) then S = S'
        update best solution found S*: if F(S) < F(S*) then S* = S
end-while
return best solution S*
```

Figure 2: Pseudo-code for the Node-Ranking Function algorithm

5.4. Preliminary computational results

Being a "new" problem, there are no "off-the-shelf" test instances available in the literature to be used for benchmarking. 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 (Haibing and Lim 2001) and the Dial-A-Ride-Problem (DARP) (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. For the computational assessment of the developed approach we decide to randomly generate a set of test instances.

Computational tests were done using an Intel Core Duo running at 1,67GHz, 2GB RAM memory, and the adjustment of the parameter that controls greediness/randomness level was done at every 100th algorithm iteration. The number of parallel threads running the algorithm is dynamically set to 8. 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. Figure 2, obtained using 50 stops and 1000 algorithm iterations shows the effect of increasing the number of passengers (requests).

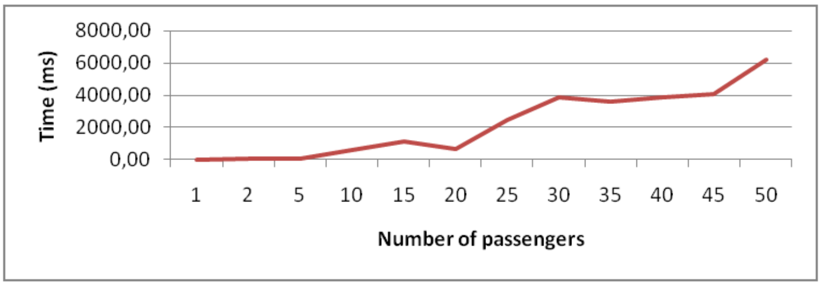

Figure 3: Effect of the number of passengers on algorithm

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. Figure 3 captures this observation for a problem with 50 stops and 20 transportation requests, gradually adding 1000 iterations to the algorithm. Each 1000 iterations takes less than 800ms.

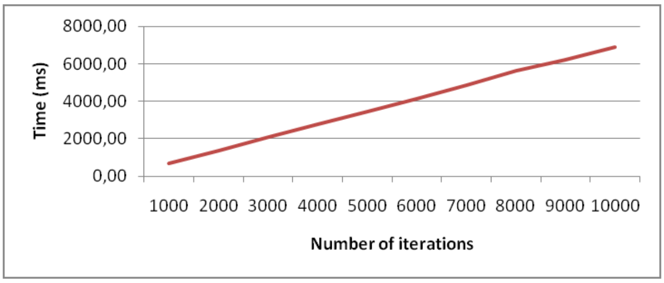

Figure 4: Number of iterations effect on the algorithm

7. DECISION SUPPORT SYSTEM

In order to "involve" the experts in the planning process, a prototype of a Decision Support System is being developed. This system integrates the multi-objective algorithmic approaches previously developed, and will be used in testing and assessing the approach. The figure illustrates the graphical user interface of the Decision Support System.

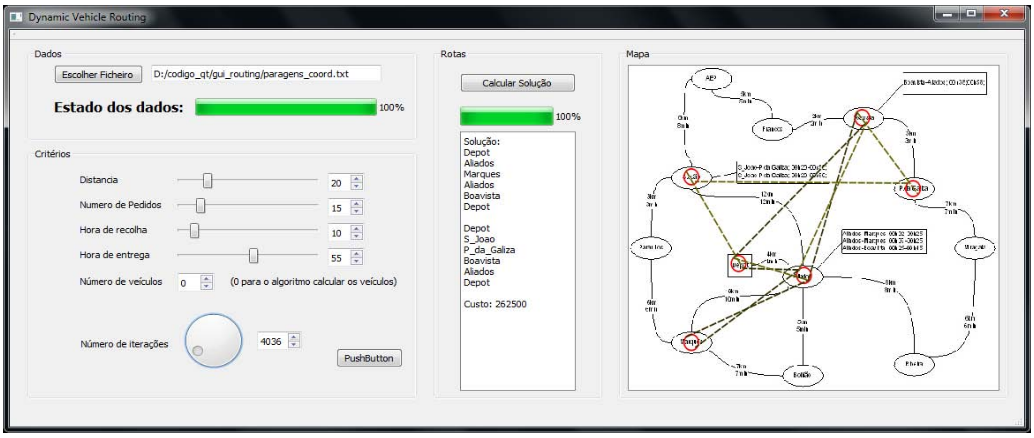

Figure 5: Decision Support System graphical user interface

The service operator can at any perform the route planning. This is the core use case of the system. The route planning can be initiated at any time by service operator request but can also be performed automatically each time a new transportation request arrives in real time. In this later case, the request feasibility must be checked. The route planning requires that the service operator specifies his perspectives/preferences, assigning weights to the different criteria: travel distance minimization, maximization of number of requests served, minimization of passenger waiting time and, finally, minimization of passenger on-board ride time. The service operator should also specify how many iterations the algorithm should perform (this, naturally, as a penalty cost in terms of algorithm run time). The routes are displayed on the area map. The routes displayed are the result set of the route planning algorithm. The total solution cost is also displayed. If the service operator wishes to do so, he/she can also check information available on the stops along the routes displayed on the map at a given time, such as number of passengers waiting at the stop and number of passenger who specified that stop as their destination.

The commuter intending to use the service must specify a request according to his transportation needs using a request client subsystem. To do so, he should define the origin, the destination, the pickup time and, finally, delivery time according to his needs. This request is checked for feasibility and, afterwards, the commuter will be noticed about the result of this feasibility test and also on the proposed pickup and delivery times.

The Decision Support System has a client-server logic architecture, based on the Three Tier Distribution Architecture pattern (Hirschfeld 1996), with a three-tier server and a thin client. This pattern is used to structure the distribution of application functionality between distributed processing contexts, in order to optimize the usage of components and resources. The architecture allows for loosely coupled clients for transportations requests to be developed, promoting interoperability between different technologies. I.e., several different technologies can be used for developing the clients: a web service, a desktop software, a mobile phone application, and so on. The following figure shows a preliminary interface of the Mobile Reservations client prototype. It was capture directly from a Nokia E71 mobile phone.

Figure 6: Mobile Reservations client interface

This is the application that the commuters use to make transportations requests using mobile phone. The commuter specifies the origin, the destination, the pickup time and the delivery time according to his needs. Note that this is a remote mobile client, so the commuter can be anywhere in the world (as long as he has a data plan or access to a Wi-Fi spot) but, as this is a preliminary prototype, he also should specify the location where the server (Decision Support System) is running – in a final product this “location” can be the name of the DRT service the transport operator offers.

8. 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, and that is why heuristic techniques are often used. Besides, in a multiple criteria decision analysis the "optimal" solution is in general meaningless because it is impossible to satisfy all (usually conflicting) objectives simultaneously.

We have presented an innovative approach for DRT services – here referred as Dynamic Vehicle Routing for Demand Responsive Transportation (DVRDRT). From a survey of 24 European DRT services, currently there is no DRT service operating with the degree of dynamism and flexibility of the DRT approach presented in this document. We aim at designing and developing a general modeling framework for planning and managing services for transportation of this type. A further subsidiary aim is to find a set of efficient solutions hopefully close to the Pareto front using efficient, customizable multi-objective algorithmic approaches, to be later embedded in a Decision Support System (DSS).

The approach proposed in this work seems to be a powerful and flexible tool to model quite different DRT services. The Parallel Reactive GRASP based 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 somehow "changing" the neighborhood 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

When designing a DRT service, it is not only important to be able to solve the model in an efficient way, but also understand how different ways of operating the service affect customers and operators. Simulation can be used as a tool to study how different ways of operating the service affect its performance and efficiency in a given scenario. Work is now being pursued in embedding such a tool in a broader Decision Support System for providing a set of efficient solutions and thus improving decision-making processes.

REFERÊNCIAS BIBLIOGRÁFICAS

Branke, J., K. Deb, et al., Eds. (2008). *Multiobjective Optimization: Interactive and Evolutionary Approaches*, Springer-Verlag.

Cordeau, J.-F. (2006). "A Branch-and-Cut Algorithm for the Dial-a-Ride Problem." *Oper. Res.* **54**(3): 573-586.

Cordeau, J.-F., G. Laporte, et al. (2007). Chapter 6 Vehicle Routing. *Handbooks in Operations Research and Management Science*, Elsevier. **Volume 14:** 367-428.

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

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

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

Fisher, M., T. L. M. C. L. M. M.O. Ball, et al. (1995). Chapter 1 Vehicle routing. *Handbooks in Operations Research and Management Science*, Elsevier. **Volume 8:** 1-33.

Gilbert, L. and J. F. Cordeau. "VRP data." Retrieved 04, 2010, from http://neumann.hec.ca/chairedistributique/data/.

Haibing, L. and A. Lim (2001). *A Metaheuristic for the Pickup and Delivery Problem with Time Windows.*

Hirschfeld, R. (1996). *Three-Tier Distribution Architecture*. PLoP '96, Allerton Park, Illinois.

Kindervater, G. A. P. and M. W. P. Savelsbergh (1997). Vehicle routing: handling edge exchanges. *Local Search in Combinatorial Optimization*. E. Aarts and J. K. Lenstra. Chichester, John Wiley & Sons**:** 311–336.

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

Lenstra, J. K. and A. H. G. R. Kan (1981). "Complexity of vehicle routing and scheduling problems." *Networks* **11**(2): 221-227.

Machado, P., J. Tavares, et al. (2002). *Vehicle Routing Problem: Doing It The Evolutionary Way*. GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference, Morgan Kaufmann Publishers.

Madsen, O., H. Ravn, et al. (1995). "A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives." *Annals of Operations Research* **60**(1): 193-208.

Nelson, J. D. (2004). *Demand responsive transport services towards the flexible mobility agency.* Rome, ENEA.

Paquette, J., J.-F. Cordeau, et al. "Quality of service in dial-a-ride operations." *Computers & Industrial Engineering* **In Press, Corrected Proof**.

Psaraftis, H. N. (1983). "k-Interchange procedures for local search in a precedence-constrained routing problem." *European Journal of Operational Research* **13**(4): 391-402.

Psaraftis, H. N. (1995). "Dynamic vehicle routing: Status and prospects." *Annals of Operations Research* **61**(1): 143-164.

Resende, M. and C. Ribeiro (2003). Greedy Randomized Adaptive Search Procedures. *Handbook of Metaheuristics*. F. G. a. G. A. Kochenberger, Springer New York. **57:** 219-249.

Solomon, M. M. (1987). "Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints." *Operations Research* **35**(2): 254-265.

Rui Jorge Gomes (dgt08009@fe.up.pt)
Jorge Pinho de Sousa (jsousa@fe.up.pt)
Teresa Galvão Dias (tgalvao@fe.up.pt)
Faculdade de Engenharia da Universidade do Porto
Rua Dr. Roberto Frias, s/n 4200-465 Porto, Portugal