Solving the on-line Dial-a-Ride Problem

Roberto Wolfler Calvo
LOSI - Université de Technologie de Troyes
e-mail: roberto.wolfler@utt.fr

Alberto Colorni, Alessio Soldano, Fabiana Valandro
DEI - Politecnico di Milano
e-mail: colorni@elet.polimi.it

Roberto Cordone
DTI - Università degli Studi di Milano
e-mail: cordone@dti.unimi.it

Abstract

The Dial-a-Ride is an emerging transport system, in which a fleet of vehicles, without fixed routes and schedules, carries people from the desired pickup point to the desired delivery point, during a pre-specified time interval. It can be modelled as an $\mathcal{NP}$-hard routing and scheduling problem, with a suitable mixed integer programming formulation. The exact approaches to this problem are insufficient for real-life problems: time dependent network, requests received on-line, different objective functions. In this paper an algorithm to solve the on-line DaR problem is proposed. It inserts instantaneously a new request and then starts an off-line optimisation phase based on the solution of an assignment problem inserted in a granular tabu search method. The algorithm, tested on instances created ad hoc using the network of Milan, prove to be fast and effective. Moreover, the same algorithm proved to be effective in solving the off-line version too.

1 Introduction

The Dial-a-Ride (*DaR*) system concerns the management of a fleet of vehicles in order to satisfy transport demands. The customers demand the service in calling a central unit and in specifying: the desired pick-up point, the delivery point (respectively, *origin* and *destination*), the number of passengers and some limitation on the service time (e. g., the earliest departure time). Such transportation system are *demand-responsive*: the routes and schedules of the vehicles change dynamically on the basis of the actual requests of the users. By better exploiting vehicle capacity, they offer the comfort and flexibility of private cars and taxies at a lower cost. The *DaR* is suited to service sparsely populated areas, or densely populated areas during weak demand periods or special classes of passengers with specific requirements (elderly, disabled).

Several models of the *DaR* service have been proposed in the literature: with or without time windows, with a fixed or unlimited fleet of vehicles, and so on. In the "static" *DaR*, the customers ask for service in advance and the plan is made before the system starts; in the "dynamic" *DaR*, the customers can call during the service time (see [2]) and the solution is updated on-line. Different objective functions have been taken into account: to minimize the number of vehicles used or the total travel time, to maximize the number of customers served or the level of service provided to the user. This paper addresses the dynamic *DaR* problem with time windows and a fleet of fixed size. Each request corresponds to a single passenger, and the objective function maximizes firstly the number of customers served, then it maximizes the level of service provided on average to the customers.

The *DaR* is $\mathcal{NP}$*-hard in the strong sense*, as it generalizes the *Pickup and Delivery Problem with Time Windows* (*PDPTW*). Nevertheless, exact algorithms for the single-vehicle case have been developed by Psaraftis [3, 4], Desrosiers et al. [5], Sexton and Bodin [6]. An exact approach for the multi-vehicle case has been proposed by Dumas et al. [7]. However, heuristics are needed for dealing with the dynamic problem. As for the heuristics, parallel insertion algorithms have been presented by Jaw et al. [8] and Madsen et al. [9]. A heuristic for the transport of handicapped people with a homogeneous fleet of vehicles has been proposed by Ioachim et al. [10] and Toth and Vigo [11]. Recently Cordeau and Laporte [1] proposed a tabu search to solve a static DAR with an homogeneous fleet of vehicles. For a complete survey on *PDPTW* and related problems, see [7] and [2]. For a recent overview on the DaR Problem see Cordeau and Laporte [12].

In this paper a very fast and effective heuristic to solve the on-line DaR is described. The on-line version is characterized by the time dependent travel time and by the dynamic arrival of the requests. Moreover, in the considered real-life case study, the requests have to be accepted or rejected in real time without any additional phone call. Thus, the heuristic is based on the idea of inserting each request with a simple and straightforward insertion procedure. Then, during the interval between two successive requests, the solution is optimized in order to create the greatest space for future customers. The improvement phase is based on a granular tabu search. One of the interesting feature of the proposed algorithm is the method to obtain the sparse (granular) graph.

2 The problem

Let $R = \{1...n\}$ be a set of requests (customers). For each request i two nodes (i^+ and i^-) are defined: a load q_i must be taken from i^+ to i^-. Let $N^+ = \{i^+|i \in R\}$ be the set of pick up nodes and $N^- = \{i^-|i \in R\}$ denotes the set of delivery nodes. A positive amount $q_{i^+} = q_i$ is associated to the pick up node, a negative amount $q_{i^-} = -q_i$ to the delivery node. A time window is also associated to each node, both to a pick up node $[e_{i^+}, l_{i^+}]$ and to a delivery node $[e_{i^-}, l_{i^-}]$. The fleet of vehicles is denoted as M; all vehicles have the same capacity Q and time window $[e_0, l_0]$.

Let $G(N, A)$ be a directed graph, whose set of vertices is defined as $N = N^+ \cup N^- \cup \{0\}$, where node 0 represents the vehicle depot. The set of arcs A is defined as $A = \{(i, j) : i, j \in N, i \neq j\}$ and each arc $(i, j) \in A$ has associated a distance $d_{i,j}$, a travel time $t_{i,j}$ and a cost function $c_{i,j}$. Another set $E = \{(i, j) : i, j \in N^+ \cup N^-, i \neq j\}$ represents the subset of arcs whose extremes are customer nodes. The problem is to find a set of routes starting and ending at the depot, such that all the customers are satisfied and the pick up node of each customer is visited before the delivery node. Moreover, the solution should be feasible with respect to the capacity constraints and the time window constraints. The variables $x^m_{i,j}$ are equal to 1 if vehicle m uses arc $(i, j) \in A$ and equal to 0 otherwise; p_i represents the departure time from node $i \in N^+ \cup N^-$; y_i is the load of the vehicle leaving node i.

$$\max \sum_{i \in N^+} \alpha_i \sum_{m \in M} \sum_{j \in N} x^m_{i,j} + \sum_{i \in N^-} LoS_i \tag{1}$$

s.t.

$$\sum_{m\in M}\sum_{j\in N} x^m_{i,j} \le 1 \qquad \forall i \in N^+ \tag{2}$$

$$\sum_{j\in N} x^m_{i,j} - \sum_{j\in N} x^m_{j,i} = 0 \qquad \forall m \in M, \quad \forall i \in N^+ \cup N^- \tag{3}$$

$$\sum_{j\in N} x^m_{i^+,j} - \sum_{j\in N} x^m_{i^-,j} = 0 \qquad \forall m \in M, \quad \forall (i^+, i^-) \in N^+ \cup N^- \tag{4}$$

$$x^m_{i,j}(y_i + q_j = y_j) \qquad \forall m \in M, \quad \forall\,(i,j) \in E \tag{5}$$

$$q_i \le y_i \le Q \qquad \forall i \in N^+ \tag{6}$$

$$0 \le y_i \le Q - q_i \qquad \forall i \in N^- \tag{7}$$

$$x^m_{i,j}(p_i + t_{i,j} \le p_j) \qquad \forall m \in M, \quad \forall\,(i,j) \in E \tag{8}$$

$$e_i \le p_i \le l_i \qquad \forall i \in N \tag{9}$$

$$p_{i^+} + t_{i^+,i^-} \le p_{i^-} \qquad \forall i = \left(i^+, i^-\right) \in N^+ \cup N^- \tag{10}$$

$$x^m_{i,j} \in \{0,1\} \qquad \forall m \in M, \quad \forall\,(i,j) \in A \tag{11}$$

Equation (1) represents the two-level objective function addressed in this paper. The first term is the number of serviced customers. The second term represents the average level of service over the set of serviced customers. The quantity α is high enough to guarantee that maximizing the number of serviced customer is the main objective. Moreover, it is important to stress that the number of vehicles is fixed and that even finding a feasible solution is itself a $NP-complete$ problem. A formal definition of the level of service is given at the end of the constraints description.

The first three groups of constraints ensure that each customer is serviced by at most one vehicle. Indeed, constraints (2) make sure that, at most, one vehicle exits from each origin node i^+, constraints (3) impose that the number of vehicles entering and exiting each node be the same, and constraints (4) that the same vehicle, if any, visits the pickup and the delivery node. Constraints (5), (6) and (7) ensure the feasibility of the loads. The number of passengers in a given vehicle varies according to the number of people boarding it or getting off it. The maximum capacity of the vehicle cannot be exceeded. The last three classes of constraints impose the feasibility of the schedule. Constraints (8) represent the compatibility requirements between routes and schedules. Constraints (9) ensure that the departure time takes place during the time window: when the vehicle arrives at node i before e_i the driver must wait; it is unfeasible to arrive at node i after l_i. Finally, constraints (10) imply that for each trip the delivery node is visited after the pickup node. Constraints (8) and constraints (5) can be linearized respectively as $M(1 - x^m_{ij}) \ge p_i + t_{ij} - p_j$ and $M(1 - x^m_{ij}) \ge y_i + q_j - y_j$. The former equations are a generalization of the classical *TSP subtour* elimination constraints proposed by Miller, Tucker and Zemlin [13].

To complete the formulation, it is necessary to write explicitly the level of service. It can be defined in many different ways, but a reasonable measure of the quality of service perceived by the customers (*level of service*, *LoS*) is given by the difference between the service time offered by the DaR system for the trip and the minimum time the customer would need to go from the origin to the destination. More precisely, the *LoS* for customer i is defined as

$$LoS_i = (t_{wait} + \Gamma_{i^+i^-}) - t_{i^+i^-}^{min} \tag{12}$$

where t_{wait} denotes the waiting time before boarding the vehicle, $\Gamma_{i^+i^-}$ denotes the total time needed to reach destination using the DaR transportation system (sum over the sequence of nodes between i^+ and i^- of waiting time and travel time) and $t_{i^+i^-}^{min}$ denotes the minimum time needed to go from the origin to the destination. Obviously $LoS_i \geq 0$ and this objective function must be minimized to optimize the service quality.

3 The algorithm

The algorithm is based on a fast insertion procedure necessary to answer customer in real time. Then, an off-line optimisation phase starts in order to improve the solution and to create the largest space for inserting future calls. The off-line optimisation procedure is based on a granular tabu search algorithm. The granular tabu search, recently proposed by Toth and Vigo in [14], is a new variant of the classical tabu search. The main components of a granular tabu search are: the neighborhood, the method to obtain the granular neighborhood, the diversification/intensification procedure, the stopping criteria and the aspiration criteria. Since it is an on-line implementation the stopping criteria is obtained as the minimal value between few seconds and the distance among two phone calls. The others components are briefly described in the following sections. Let's recall that the same components have been used to solve the static version of the problem.

3.1 The initial solution

The main characteristic of the on-line version of the problem is that the requests arrive during the service, thus the solution has to be modified in real time. Moreover, the algorithm takes into account the traffic conditions on the network. It means that the minimal path between two nodes has to be calculated in real time, when the departure time from the original node is known (see [15]). Due to this two characteristics the proposed algorithm tries to insert the new request in the existing solution with the fastest possible method: a straightforward insertion procedure. When the resulting solution is feasible the client is accepted, otherwise he/she is rejected. The method allows to answer instantaneously to the customer, but it disregards completely any optimisation phase.

Then, giving the set of vehicles and the inserted customers an off-line optimisation phase starts. First of all, a better and more useful initial solution is built. The used approach starts considering a static Dial-a-Ride problem obtained as follows. The number of vehicles is fixed and equals to those effectively used until now: at most the whole fleet. The set of customers is constituted by the customers actually inserted on the bus, but still unserved since they will be picked up in the future.

The new initial solution is obtained solving an assignment problem as described in [16]. The approach consists in building an auxiliary graph simpler and smaller than the real one. The graph has a node for each customer without any time window associated and a weighted edge (i, j) between each pair of nodes. The weight on each edge is a coefficient $\bar{p}_{ij}$ obtained in such a way that the spatial and temporal proximity between the two customers is implicitly considered. Moreover, this graph is augmented adding two nodes for each vehicle effectively used. On this resulting graph, combining vehicles and customers, an assignment problem is solved.

The assignment problem is the simpler and easier relaxation for any routing problem, but the obtained solution is normally too far from a solution of the original problem. Nevertheless, the proposed method has the advantage that thanks to $\bar{p}_{ij}$ the obtained solution is made by a set of chains of *pickup* nodes (i.e. routes of customers) starting and ending in the depot (see in [16] for details). Unfortunately some subtours could be present too, but they are few. As for the chains of clients they are still unfeasible for the original problem: only a single node for each client has been inserted and moreover without guarantying to respect the time window constraints. The solution for the DaR problem is obtained inserting all the *delivery* nodes in the "chains". After that the *subtours* are inserted. The solution obtained is a better initial feasible solution. Moreover, the assignment provides a second useful information used to accelerate the improving phase.

3.2 The granular neighborhoods

The described insertion procedures are fast, but to have an efficient and effective algorithm a fast improving phase is needed. The implemented algorithm is a Tabu Search, but with a particular focus on the computing time. In order to reduce the computing time a general and effective search intensification tool has been implemented. The method, proposed for the first time in [14], is based on using a drastically restricted neighborhood obtained from the standard one by removing the moves that involve only elements which are not likely to belong to high-quality feasible solutions. The authors called this neighborhood a *granular neighborhood* and the whole method a *granular tabu search*.

The interesting and new idea proposed in this algorithm is the method to restrict the neighborhood. Let's say that the neighborhood used is a simple one: it consists in removing a customer from a route and inserting it into another different route. Thus, given the simple neighborhood used, it is obvious that the restriction must be performed in an efficient way. The information useful to reduce the neighborhood are obtained by the reduce cost given by the assignment problem. An arc with a reduce cost equal to zero represents a measure of proximity between two customers. Indeed, the weight initially assigned to an edge connecting two customers (nodes) is obtained combining the temporal and the spatial distance among them. A reduced cost equal zero means that they are closed and when they are located in two different vehicles this suggest a possible interesting movement. Thus, the algorithm starts with a new sparse or *granular* graph composed with the set of edges whose reduced cost is equal zero. The search of a granular neighborhood considers only the moves which are generated by arcs belonging to the granular graph.

3.3 Diversification strategies

The diversification strategies used are two: one derives naturally as consequence of the granularization process. The other one is based on the idea of measuring the frequency with which arcs get in or out of the current solution. As for the former it exploits the idea of changing dynamically the structure of the sparse graph associated to the granular neighborhood. Every time the algorithm is unable to improve the solution using a certain granular graph, it is enlarged considering arcs with a positive, but still small, reduce cost. Whenever the algorithm improves the current solution the sparse graph becomes small again. The second diversification strategy penalizes each solution $\bar{s} \in N(s)$ such that $f(\bar{s}) > f(s)$ where s is the current solution. The penalty function depends on the frequency measured as number of times an edge has been used in the solution. This latest strategy has been proposed by Taillard [17]) and successfully used in many other tabu search applied to the *vehicle routing* (see for example Gendreau et al. [18] and Cordeau and Laporte 2002 [1]).

3.4 Aspiration criteria

In a TS framework, search is guided by memories. The memory, called tabu list, is used to avoid cycling. It records the most recent moves in order to forbid their reversal. However, to prevent these restrictions from being too stringent, if a change on a tabu variable leads to a better solution than the best found so far, the change is allowed *aspiration criteria*.

4 Computational results

The algorithm as been tested on a data set has been created using the complete network of Milan (about 17000 arcs and 7000 nodes) and are proposed as benchmark problems for the future. The set of random instances have been created extracting casually a pair of nodes for each customers. Each customer requiring either the arrival time or the departure time. The time is generated randomly in a time interval of 240 minutes (four hours) and in the same way it is associated to the pickup node or to the delivery node.

References

[1] J.-F. Cordeau and G. Laporte. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research B*, 37:579–594, 2002.

[2] M. W. P. Savelsbergh and M. Sol. The general pickup and delivery problem. 29(1):17–29, 1995.

[3] H. N. Psaraftis. A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. 14(2):130–154, 1980.

[4] H. N. Psaraftis. An exact algorithm for the single vehicle many-to-many dial-a-ride problem with time windows. 17(3):351–357, 1983.

[5] J. Desrosiers, Y. Dumas, and F. Soumis. A dynamic programming solution of the large-scale single-vehicle Dial-a-Ride problem with time windows. 6, 1986.

[6] T. Sexton and Y. Choi. Pick-up and delivery of partial loads sith time windows. 6:369–398, 1986.

[7] Y. Dumas, J. Desrosiers, and F. Soumis. The pickup and delivery problem with time windows. 54:7–22, 1991.

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

[9] O. B. G. Madsen, H. F. Ravn, and J. M. Rygaard. A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities and multiple objectives. 60:193–208, 1995.

[10] I. Ioachim, J. Desrosiers, Y. Dumas, M.M. Solomon, and D. Villeneuve. A request clustering algorithm for dorr-to-door handicapped transportation. 29:63–78, 1995.

[11] P. Toth and D. Vigo. Heuristic algorithms for the handicapped persons transportation problem. 31(1):60–71, 1997.

[12] J.-F. Cordeau and G. Laporte. The dial-a-ride problem (darp): Variants, modeling issues and algorithms. *4OR-Quaterly Journal of theBelgian, French and Italian Operations Research Societies*, 1:89–101, 2003.

[13] C. E. Miller, A. W. Tucker, and R. A. Zemlin. Integer programming formulations and traveling salesman problems. 7:326–329, 1960.

[14] P. Toth and D. Vigo. The granular tabu search (and its application to the vehicle routing problem). 2001.

[15] K. Sung, M.G.H. Bell, M. Seong, and S. Park. Shortest paths in a network with time-dependent flow speeds. *European Journal of Operational Research*, 121:32–39, 2000.

[16] R. Wolfler Calvo and A. Colorni. An approximation algorithm for the dial-a-ride problem. 2003.

[17] E. D. Taillard. Parallel iterative search methods for vehicle routing problem. *Networks*, 23:661–673, 1993.

[18] M. Gendreau, A. Hertz, and G. Laporte. A tabu search huristic for the vehicle routing problem. *Management science*, 13:1276–1290, 1994.