A Matheuristic for the Dial-a-Ride Problem

Roberto Wolfler Calvo$^1$ and Nora Touati-Moungla$^2$

$^1$ 99, Avenue J.-B. Clement, 93430 Villetaneuse, France
Roberto.Wolfler@lipn.univ-paris13.fr

$^2$ École polytechnique, Laboratoire d’informatique (LIX), 91128 Palaiseau Cedex France

Abstract. The Dial-a-Ride is a transport system on demand. A fleet of vehicles, without fixed routes and schedules, carries people from their pickup points to their delivery points, during a pre-specified time interval. It can be modeled as an $\mathcal{NP}$-hard routing and scheduling problem, with a suitable mixed integer programming formulation. Exact approaches to this problem are too limited to tackle real-life instances: time dependent network, requests received on-line, different objective functions. In this paper we propose an algorithm to solve the off-line Dial-a-Ride Problem (DARP), based on a Granular Tabu Search method. This algorithm was fast and effective, when tested on instances created ad hoc using the Milan network.

1 Introduction

The Dial-a-Ride (DAR) system concerns the management of a fleet of vehicles in order to satisfy transport demands. The customer requests the service by calling a central unit and specifying: the pick-up point, the delivery point (respectively, origin and destination), the number of passengers and some limitations on the service time (e.g., the earliest departure time). Such transportation system is called on demand: the routes and schedules of the vehicles change dynamically on the basis of the current requests of the users. By better exploiting vehicle capacity, they offer the comfort and flexibility of private cars and taxis at a lower cost. DAR is suited to service sparsely populated areas, 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 customer asks for service in advance and the vehicles are routed before the system starts to operate; in the “dynamic” DAR, the customer can call during the service time and the routes are updated on-line. Different objective functions have been taken into account: minimization of the number of vehicles used or the total travel time, maximization of the number of customers served or the level of service provided to the user. This paper addresses the static DAR with time windows and a fixed fleet of vehicles. Each request corresponds to a single passenger, and the objective function maximizes firstly the number of customers served, then it minimizes the number of vehicles used and finally 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) [6]. Nevertheless, exact
algorithms for the multi-vehicle case [16] have been developed. However, heuristics and matheuristics are needed [3,8,9,13], while exact methods are useless, for dealing with the dynamic problem with time dependent network.

This paper describes a fast and effective matheuristic for improving the feasible initial solution obtained at the end of the algorithm described in [16] to solve the off-line DARP. The algorithm is based on a Granular Tabu Search where the proposed procedure for obtaining the granular graph is different from the one proposed in [14], since it is based on the reduced costs matrix obtained by solving a simpler, but useful, subproblem. The algorithm transforms the original graph of nodes (i.e. pick-up and delivery nodes for each customer) into a graph of customers, which is smaller than the original one. A value is assigned to each arc of the auxiliary graph, measuring the distance between each pair of customers. Then, a classical assignment problem is defined and solved on this graph. The solution obtained gives several information on the instance. The most useful one is the reduced costs matrix which shows how close customers are to each other (a short edge has a reduced cost equal zero). The set of arcs is therefore ordered by increasing value of the reduced costs. The proposed Granular Tabu Search exploits this information to guide the local search.

The paper is organized as follows. The problem is described in the next section together with its mixed linear programming formulation. In Section 3 the general Granular Tabu Search approach is described. Section 4 explains the problems addressed for applying the Granular Tabu Search to the DARP and how they have been solved. The whole algorithm used is described in Section 5. Computational results and conclusions close the paper.

2 The Problem

Let \( R = \{1...n\} \) be a set of requests (customers). For each request \( i \) two nodes \((i^+, i^-)\) 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 \( N^- = \{i^- | i \in R\} \) the set of delivery nodes. A positive amount \( q_{i^+} = q_i \) is associated with the pick up node, a negative amount \( q_{i^-} = -q_i \) with the delivery node. A time window is also associated with each node (i.e. pick up node \([e_j, l_j]\) and delivery node \([e_i, l_i]\)). The fleet of vehicles is denoted as \( V \); 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 is the depot. The set of arcs \( A \) is defined as \( A = \{(i,j) : i, j \in N, i \neq j\} \) with a distance \( d_{i,j} \) or a travel time \( t_{i,j} \) assigned to each arc \((i,j) \in A\). 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 consists in finding a set of routes starting and ending at the depot, such that an objective function is optimized. The pick up node of each customer must be visited before the delivery node in the same route. Capacity and time window constraints must also be respected. The variables \( x_{i,j}^v \) are equal to 1 if vehicle \( v \) 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 \( i \).

\[
\max z(P) = \max \left( \alpha_1 \sum_{i \in N^+} \sum_{v \in V} \sum_{j \in N} x_{i,j}^v - \alpha_2 \sum_{v \in V} \sum_{j \in N^+} x_{0,j}^v - \alpha_3 \sum_{i \in N^-} S_i \right) \quad (1)
\]