Vehicle Routing for the Home Delivery of Perishable Products

Nabila Azi*† Michel Gendreau*† Jean-Yves Potvin*†

*Centre de recherche sur les transports
Université de Montréal
C.P. 6128, Succursale Centre-Ville, Montréal H3C 3J7 Canada
{azn,michelg}@crt.umontreal.ca

†Département d'informatique et de recherche opérationnelle
Université de Montréal
C.P. 6128, Succursale Centre-Ville, Montréal H3C 3J7 Canada
potvin@iro.umontreal.ca

1 Introduction

We consider a variant of the standard vehicle routing problem where the same vehicle is assigned to several routes during a given planning period. This problem has received little attention in the literature (apart from [3, 5]), in spite of its importance in practice. For example, the home delivery of perishable goods, like food, implies that routes are of short duration and must be combined to form a complete working day. We believe that this type of problem will become increasingly important in the future with the advent of electronic services, like e-groceries, where customers can order goods through the Internet and get them delivered at home [1].

2 Problem statement

The problem that we consider can be stated as follows. We have a single vehicle of capacity Q delivering perishable goods from a depot node to a set of customer nodes $N = \{1, 2, ..., n\}$ in a complete directed graph with arc set A. A distance d_{ij} and a travel time t_{ij} are associated with every arc $(i, j) \in A$. Each customer $i \in N$ is characterized by a demand q_i, a revenue p_i, a service or dwell time s_i and a time window $[a_i, b_i]$, where a_i is the earliest time to begin service and b_i is the latest time. Hence, the vehicle must wait if it arrives at customer i before time a_i. The working day of a vehicle is made of a set of routes $K = \{1, 2, ..., k\}$ where each route starts and ends at the depot (some of these routes might be empty). We assume, without loss of generality, that the routes are served in the order 1, 2,..., k. The depot is denoted by 0 or $n+1$ depending if it is the initial or terminal node of an arc, with $s_0 = s_{n+1} = 0$, $q_0 = q_{n+1} = 0$; the symbol N^+ is used for $N \bigcup \{0, n+1\}$ and A^+ for $A \bigcup \{(0, n+1)\}$, where $(0, n+1)$ is a dummy arc with distance $d_{0(n+1)} = 0$ and travel time $t_{0(n+1)} = 0$. Due to the presence of perishable items, every customer in a route must be served before a given deadline associated with that route. The latter is defined by adding a constant t_{max} to the route start time. Also, a set up or vehicle loading time σ^r is associated with each route $r \in K$. The objective is to minimize the total distance traveled to serve all customers while satisfying the capacity, time window and deadline constraints.

This problem can be formulated as follows, using M as an arbitrary large constant:

$$\min \sum_{r \in K} \sum_{(i,j) \in A} d_{ij} x^r_{ij} \tag{1}$$

subject to

$$\sum_{j \in N^+} x^r_{ij} = y^r_i, \quad i \in N,\ r \in K, \tag{2}$$

$$\sum_{r \in K} y^r_i = 1, \quad i \in N, \tag{3}$$

$$\sum_{i \in N^+} x^r_{ih} - \sum_{j \in N^+} x^r_{hj} = 0, \quad h \in N,\ r \in K, \tag{4}$$

$$\sum_{i \in N^+} x^r_{0i} = 1, \quad r \in K, \tag{5}$$

$$\sum_{i \in N^+} x^r_{i(n+1)} = 1, \quad r \in K, \tag{6}$$

$$\sum_{i \in N} q_i y^r_i \leq Q, \quad r \in K, \tag{7}$$

$$t^r_i + s_i + t_{ij} - M\left(1 - x^r_{ij}\right) \leq t^r_j, \quad (i,j) \in A^+,\ r \in K, \tag{8}$$

$$a_i y^r_i \leq t^r_i \leq b_i y^r_i, \quad i \in N,\ r \in K, \tag{9}$$

$$t^1_0 \geq \sigma^1, \tag{10}$$

$$t^r_{n+1} + \sigma^{r+1} \leq t^{r+1}_0, \quad r = 1, ..., k-1, \tag{11}$$

$$\sigma^r = \sum_{i \in N} s_i y^r_i, \quad r \in K, \tag{12}$$

$$t^r_i \leq t^r_0 + t_{max}, \quad i \in N,\ r \in K, \tag{13}$$

$$x^r_{ij} \quad binary, \quad (i,j) \in A^+,\ r \in K, \tag{14}$$

$$y^r_i \quad binary, \quad i \in N,\ r \in K, \tag{15}$$

where

- x^r_{ij} is 1 if arc $(i,j) \in A^+$ is in route r, 0 otherwise; note that $x^r_{0(n+1)}$ is 1 if route r is empty;

- y_i^r is 1 if customer i is in route r, 0 otherwise;
- t_i^r is the time of beginning of service of customer i in route r;
- t_0^r is the start time of route r;
- t_{n+1}^r is the end time of route r.

In this formulation, equation (3) states that every customer should be visited exactly once. Equations (4), (5) and (6) are flow conservation constraints that describe the path followed by the vehicle. Equation (7) states that the total demand on a route should not exceed vehicle capacity. Equations (8), (9), (10) and (11) ensure feasibility of the time schedule. Equation (12) defines the vehicle loading time as the sum of service times of all customers in a route (but could be defined otherwise). Finally, equation (13) corresponds to the deadline constraint for the service at a customer. Note that equation (9) forces the t_i^r variables to 0 when customer i is not in route r. Consequently, equation (13) is automatically satisfied in this case.

In practice, it might not be possible to serve all customers with the vehicle due to time window constraints. An alternative objective could thus be:

$$\min \sum_{r \in K} \sum_{(i,j) \in A} d_{ij} x_{ij}^r - \alpha \sum_{r \in K} \sum_{i \in N} p_i y_i^r,$$

where α is a weighting parameter for the revenue, which should be large enough to provide an incentive to visit customers. The equality sign in equation (3) of the model is also replaced by a $\leq$ sign.

3 Problem-solving methodology

The problem introduced in the previous section is addressed via a problem-solving approach that exploits an elementary shortest path algorithm with resource constraints [2], noted FDGG in the following, which can be applied to graphs with negative cycles. This label correcting algorithm maintains, for each partial path from an origin node s to some node i, a label that contains the path length, a vector of resources (e.g., time, capacity) consumed by the path, and a trace of all visited nodes. If there are two distinct paths leading from s to i, then one dominates the other if (1) it is of shorter length, (2) it consumes less resources, for every type of resources considered and (3) the nodes that it contains form a subset of the nodes visited by the other (we might have equalities between the two paths for some of these conditions, but at least one must strictly hold). As stated in [2], only non-dominated paths can lead to the optimal solution. Consequently, partial paths can be eliminated through the dominance criterion.

Our problem-solving methodology, based on the above algorithm, is divided into two phases:

Phase 1.

In this first phase, all feasible routes are identified. As the deadline constraint is quite restrictive in practice, even the largest feasible routes will contain only a few customers and a pure enumerative approach is viable. Through a suitable adaptation of the FDGG algorithm, however, it is possible to obtain the same enumeration, but with less computational effort. To this end, the distance d_{ij} is replaced by $d_{ij} - \alpha$ on every arc of the graph, with $\alpha > max_{(i,j) \in A}\ d_{ij}$. As it is always beneficial to extend the path with a new arc, the corresponding distance being negative, this modification forces the algorithm to visit customers (otherwise, the optimal solution would be to stay at the depot).

To generate all feasible routes, condition (3) in the dominance criterion is modified and now states that the nodes visited in one path should be the same as those in the other path. With this modification, a path is discarded when another leads to the same node, going through the same intermediary nodes (although in a different order), and is of smaller length and consumes less resources. In this way, the non-dominated routes obtained at the end of the FDGG algorithm correspond to all feasible routes. These are the building blocks used in Phase 2 to produce a solution to our problem, as it is explained below.

Phase 2.

The same FDGG algorithm is used again to create a working day for the vehicle. Here, the shortest path algorithm is applied on a transformed graph where the nodes are the feasible routes generated in Phase 1, plus an artificial starting node and an artificial ending node. An arc is present between two nodes r and r' in this graph, if it is feasible to serve route r' after route r. The distance associated with arc (r, r') is:

$$\sum_{(i,j) \in A} d_{ij} x_{ij}^{r'} - \alpha' \sum_{i \in N} p_i y_i^{r'},$$

which is the total distance of route r' minus a weighting parameter α' times the revenue of route r'. If we assume that the triangle inequality holds, negative values are obtained by setting $\alpha' > 2\ max_{i \in N}\ d_{0i}$. The shortest path algorithm, applied from the artificial starting node to the artificial ending node, will thus consider the inclusion of a route if it is feasible to do so.

This algorithm is currently being tested on problem instances derived from Solomon's benchmark problems [4], using more or less restrictive deadline constraints for the service at customers.

References

[1] A.M. Campbell and M. Savelsbergh, "Decision Support for Consumer Direct Grocery Initiatives", Report TLI-02-09, Georgia Institute of Technology, Atlanta, GA, 2002.

[2] D. Feillet, P. Dejax, M. Gendreau and C. Gueguen, "An Exact Algorithm for the Elementary Shortest Path Problem with Resource Constraints: Application to Some Vehicle Routing Problems", Technical Report CRT-2000-15, Centre de recherche sur les transports, Montréal, Canada, 2000.

[3] B. Fleischmann, "The Vehicle Routing Problem with Multiple Use of Vehicles", Working Paper, Fachbereich Wirtschaftswissenschaften, Universität Hamburg, Hamburg, Germany, 1990.

[4] M.M. Solomon, "Algorithms for the Vehicle Routing and Scheduling Problem with Time Window Constraints", *Operations Research* 35, 254–265 (1987).

[5] É. D. Taillard, G. Laporte and M. Gendreau, "Vehicle Routeing with Multiple Use of Vehicles", *Journal of the Operational Research Society* 47, 1065–1070 (1996).