Generalizations of the vehicle routing problem with time windows

H.Hashimoto, T.Ibaraki, S.Imahori and M.Yagiura

Abstract

The *vehicle routing problem with time windows* (VRPTW) is the problem of minimizing the total travel distance of a number of vehicles, under capacity and time window constraints, where every customer must be visited exactly once by a vehicle. We consider two generalizations of the VRPTW and propose local search algorithms. The first one treats traveling times as variables and introduce cost functions on them. This enables us to shorten the traveling times by adding some cost, thus introducing a new dimension to consider flexible solutions. The other allows both traveling times and traveling costs being time-dependent, so that it can treat time-dependent situations such as rush-hour traffic jam. In our local search algorithms for these problems, after fixing a route of each vehicle, we must compute an optimal time schedule of each route. We investigate the complexity of this subproblem and show that it can be efficiently solved by dynamic programming under some conditions.

1 Introduction

The *vehicle routing problem with time windows* (VRPTW) is the problem of minimizing the total travel distance of a number of vehicles, under capacity and time window constraints, where every customer must be visited exactly once by a vehicle. The capacity constraint signifies that the total load on a route cannot exceed the capacity of the assigned vehicle. The time window constraint signifies that each vehicle must start the service at each customer in the period specified by the customer. The VRPTW has a wide range of applications such as bank deliveries, postal deliveries, school bus routing and so on, and it has been a subject of intensive research focused mainly on heuristic and metaheuristic approaches. In this problem, even just finding a feasible schedule with a given number of vehicles is known to be NP-complete. It may not be reasonable to restrict the search only within the feasible region, especially when the constraints are tight. Moreover, in real-world situations, time window and capacity constraints can often be violated to some extent. The violation of constraints is usually penalized and added to the objective function [3, 5]. Ibaraki et al. [3] proposed the *vehicle routing problem with general time windows* (VRPGTW), where the time window constraints are treated as cost functions that can be non-convex and/or discontinuous as long as it is piecewise linear.

We proposed two generalizations of the VRPGTW in [1, 2], which are also generalizations of the VRPTW. This extended abstract is a short summary of our recent results in [1, 2]. The first one is the vehicle routing problem with flexible time windows and traveling times. In this generalization, we treat traveling times as variables. In practice, traveling times can be changed with some cost (e.g., the traveling time can be shortened by paying the turnpike toll). Our goal is to find such a flexible solution. The other one is a time-dependent case of the VRPGTW. In real situations, traveling time between customers is often dependent on the leaving time, and it cannot be treated as a constant in such cases (e.g., rush-hour traffic jam). In this generalization, traveling times and costs are dependent on the leaving time. Recently Ichoua et al. [4] proposed a similar generalization of the VRPTW, where each customer has only one time window and the objective is to minimize the total traveling time. Note that, in our generalization, each traveling cost can be different from the traveling time.

Let $G = (V, E)$ be a complete directed graph with vertex set $V = \{0, 1, \ldots, n\}$ and edge set $E = \{(i, j) \mid i, j \in V, i \neq j\}$, and $M = \{1, 2, \ldots, m\}$ be a vehicle set. Let σ_k denote the route traveled by vehicle k, where $\sigma_k(h)$ denotes the hth customer in σ_k, and let

$$\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \ldots, \sigma_m).$$

Note that each customer i is included in exactly one route σ_k, and is visited by the vehicle k exactly once. We denote by n_k the number of customers in σ_k. We define $\sigma_k(0) = 0$ and $\sigma_k(n_k+1) = 0$ for all k (i.e., each vehicle $k \in M$ leaves the depot and comes back to the depot).

In both generalizations, our objective function is a weighted sum of the total time window cost for customers, the total traveling cost, and the total capacity excess for vehicles. If vehicle routes $\boldsymbol{\sigma}$ are fixed, the problem is separated into m scheduling problems of finding the optimal start times for each σ_k. Let $cost(\sigma_k)$ be the optimal cost of σ_k (i.e., when the schedule of vehicle k on σ_k is optimized). Then, the problem can be represented as:

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{k=1}^{m} cost(\sigma_k) \\ \text{subject to} & (\sigma_1, \sigma_2, \ldots, \sigma_m) \in F, \end{array}$$

where F is the set of the feasible route sets (i.e, every customer is visited exactly once by a vehicle). Hence our algorithm searches $\boldsymbol{\sigma}$ by local search and solves the corresponding m scheduling problems for each $\boldsymbol{\sigma}$ generated during the search. In the following sections, we discuss two scheduling problems corresponding to the generalizations, respectively. How to search σ_k will be discussed in Section 4.

For convenience, in Sections 2 and 3, we assume that vehicle k visits customers $1, 2, \ldots, n_k$ in this order and let customer $n_k + 1$ represent the arrival at the depot.

2 The vehicle routing problem with flexible time windows and traveling times

Here we formulate the vehicle routing problem with flexible time window and traveling time constraints (see the details in [1]). We treat the traveling time as the difference between the start times of services at two consecutive customers, and introduce its cost function.

Each customer i, each vehicle k and each edge $(i, j) \in E$ are associated with:

1. a fixed quantity a_i (≥ 0) of goods to be delivered to i,
2. a time window cost function $p_i(t)$ of the start time t of the service at i ($p_0(t)$ is the time window cost function of the arrival time t at the depot),
3. a capacity u_k (≥ 0) of k,
4. a traveling time cost function $q_{ij}(t)$ of the traveling time t from i to j.

We assume $a_0 = 0$ without loss of generality. We assume that each time window cost function $p_i(t)$ is nonnegative, piecewise linear and lower semicontinuous (i.e., $p_i(t) \leq \lim_{\varepsilon \to 0} \min\{p_i(t + \varepsilon), p_i(t - \varepsilon)\}$ at every discontinuous point t). Note that $p_i(t)$ can be non-convex and discontinuous as long as it satisfies the above conditions. We also assume $p_i(t) = +\infty$ for $t < 0$ so that the start time t of the service is nonnegative. Similarly, we assume that each traveling time cost function $q_{ij}(t)$ is nonnegative, piecewise linear and lower semicontinuous. We also assume $q_{ij}(t) = +\infty$ for $t < 0$ so that the traveling time t between customers is nonnegative. These assumptions ensure the existence of an optimal solution. We further assume that the linear pieces of each piecewise linear function are given explicitly.

We now consider the problem of determining the times to start services at customers in a given route σ_k so that the total of time window and traveling time costs is minimized. The scheduling problem is formulated as follows:

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{h=1}^{n_k+1} p_h(s_h) + \sum_{h=1}^{n_k+1} q_{h-1,h}(s_h - s_{h-1}) \\ \text{subject to} & s_0 = 0, \end{array}$$

where s_h is the start time of service of customer h.

We proved that this problem is NP-hard in general. Under the assumption that each breakpoint of the input functions is integer (the restricted problem is still NP-hard), we proposed a dynamic programming algorithm of pseudo polynomial time. We showed that the same dynamic programming can be implemented so that, if each traveling time cost function is convex (but the time window cost functions can be general), it runs in $O\left(n_k \Delta_1(\sigma_k)\right)$ time, where

$$\Delta_1(\sigma_k) = \sum_{h=1}^{n_k+1} \delta(p_h) + \left(\sum_{h=1}^{n_k+1} \widehat{\delta}(p_h)\right)\left(\sum_{h=1}^{n_k+1} \delta(q_{h-1,h})\right),$$

$\delta(\cdot)$ is the number of linear pieces of the argument function, and $\widehat{\delta}(\cdot)$ is the number of convex intervals of the argument function. Note that $\Delta_1(\sigma_k)$ is of polynomial order of the input size.

3 The time-dependent vehicle routing problem with time windows

In this section, we formulate the time-dependent vehicle routing problem with time windows (see the details in [2]). We introduce traveling time and cost functions between each customer, whose values are dependent on the start time of traveling.

In this problem, each customer i, each vehicle k and each edge $(i, j) \in E$ are associated with:

1. a fixed quantity a_i (≥ 0) of goods to be delivered to i,
2. a time window cost function $p_i(t)$ of the start time t of the service at i ($p_0(t)$ is the time window cost function of the arrival time t at the depot),
3. a capacity u_k (≥ 0) of k,
4. a time-dependent traveling time function $\lambda_{ij}(t)$ of the start time t of traveling from i to j,
5. a time-dependent traveling cost function $r_{ij}(t)$ of the start time t of traveling from i to j.

We assume $a_0 = 0$ without loss of generality. We assume that each time window cost functions $p_i(t)$ is nonnegative, piecewise linear and lower semicontinuous as in Section 2, and that each traveling cost function $r_{ij}(t)$ satisfies the same conditions as $p_i(t)$. We also assume that each traveling time function $\lambda_{ij}(t)$ is nonnegative, piecewise linear and continuous and that $\lambda_{ij}(t)$ satisfies $t \leq t' \Rightarrow t+\lambda_{ij}(t) \leq t'+\lambda_{ij}(t')$ (called the FIFO condition in this paper). In our problem, the linear pieces of each piecewise linear function are given explicitly. Then the scheduling problem is formulated as follows:

$$\begin{array}{lll} \text{minimize} & \displaystyle\sum_{h=1}^{n_k+1} p_h(s_h) + \sum_{h=0}^{n_k} r_{h,h+1}(t_h) & \\ \text{subject to} & s_h \leq t_h, & 1 \leq h \leq n_k \\ & t_h + \lambda_{h,h+1}(t_h) \leq s_{h+1}, & 0 \leq h \leq n_k, \end{array}$$

where s_h is the start time of service of customer h and t_h is the start time of traveling from customer h.

We proposed a dynamic programming algorithm that solves the problem in $O(n_k \Delta_2(\sigma_k))$ time, where

$$\Delta_2(\sigma_k) = \sum_{h=1}^{n_k+1} \delta(p_{\sigma_k(h)}) + \delta(r_{\sigma_k(h-1),\sigma_k(h)}) + \delta(\lambda_{\sigma_k(h-1),\sigma_k(h)})$$

and $\delta(\cdot)$ is the number of linear pieces of the argument function. Note that $\Delta_2(\sigma_k)$ is the same as the input size of the problem. If traveling cost and time functions are constant functions (i.e., if there is no time dependency), this time complexity of the dynamic programming algorithm becomes the same as that of Ibaraki et al. [3], which deals with this special case.

In the case the FIFO condition does not hold, whether the algorithm runs in polynomial time or not, and whether the scheduling problem in this section is NP-hard or not are both open.

4 Framework of our algorithms

In this section, we describe a framework of our local search (LS) for finding good visiting orders $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \ldots, \sigma_m)$. It starts from an initial solution $\boldsymbol{\sigma}$ and repeats replacing $\boldsymbol{\sigma}$ with a better solution in its neighborhood $N(\boldsymbol{\sigma})$ until no better solution is found in $N(\boldsymbol{\sigma})$. We evaluate a solution $\boldsymbol{\sigma}$ by $\sum_{k=1}^{m} cost(\sigma_k)$. For $N(\boldsymbol{\sigma})$, we use standard neighborhoods called 2-opt*, cross exchange and Or-opt neighborhoods with slight modifications. Each neighborhood is the set of solutions which are obtained by applying its neighborhood operations to the current solution. A 2-opt* operation removes two edges from two different routes (one from each) to divide each route into two parts and exchanges the second parts of the two routes. A cross exchange operation removes two paths from two routes (one from each) of different vehicles and exchanges them. An intra-route operation removes a path and inserts it into another position of the same route from the original position. Our LS searches the above intra-route neighborhood, 2-opt* neighborhood and cross exchange neighborhood, in this order. Whenever a better solution is found, we immediately accept it (i.e., we adopt the first admissible move strategy), and resume the search from the intra-route neighborhood.

To achieve further improvement, we use the iterated local search (ILS), which iterates LS many times from those initial solutions generated by perturbing the best solution obtained by then. Moreover, in the evaluation of these neighborhood solutions, we reduce the time to compute the dynamic programming in both generalizations by using information from the previous computation. We also incorporate other ideas proposed in Ibaraki et al. [3] to make the search more efficient.

5 Computational experiments

In this section, we briefly summarize our experimental results for the two problems.

5.1 The vehicle routing problem with flexible time windows and traveling times

We conducted experiments for the benchmark instances of the VRPTW and the instances with flexible time windows and traveling times, which we generated from the former. For the VRPTW instances, our algorithm could obtain the same quality as the best known solutions for many instances. Furthermore, for the generated instances, we could obtain solutions with smaller number of vehicles or with much shorter traveling distances than the best known solutions of the original instances by allowing small violation of constraints (i.e., shortening the traveling times or breaking the time windows slightly). These violations should be acceptable in many practical applications, or at least it provides the information about feasibility bottlenecks. This kind of information could not be obtained by other standard approaches.

5.2 The time-dependent vehicle routing problem with time windows

We conducted experiments for the time-dependent vehicle routing problem with time windows, which we generated from the benchmark instances of the VRPTW. For comparison purposes, we also applied our algorithm to the instances after replacing the time-dependent traveling time with the fixed constant determined by taking the average of the traveling time in the whole periods, and only evaluated the output solution exactly (i.e., considering the time-dependency). This simulates the situation where we do not have an algorithm that can handle time-dependency during the search. We could observe that both time window costs and traveling costs obtained by considering time-dependency during the search are smaller than those without considering time-dependency during the search. The deference becomes larger as the instances become more time-dependent. This indicates the usefulness of our algorithm that can handle time-dependency.

6 Conclusion

We considered two generalizations of the vehicle routing problem with time windows and proposed local search algorithms. The first one treats traveling times as variables and introduce cost functions on them. This enables us to shorten the traveling times with some cost and to find flexible solutions. The other allows both traveling times and traveling costs to be time-dependent functions, and it can treat time-dependent situations such as rush-hour traffic jam. Both generalizations are very general, and include various problems such as parallel machine scheduling problems as their special cases. In our local search procedure, after fixing a route of each vehicle, we must compute an optimal time schedule of each route. We investigated the complexity of this subproblem and showed that it can be efficiently solved by dynamic programming under some conditions. We then confirmed through computational experiments the benefits of the proposed generalizations.

References

[1] H. Hashimoto, T. Ibaraki, S. Imahori, and M. Yagiura, The vehicle routing problem with flexible time windows and traveling times, *Discrete Applied Mathematics*, to appear.

[2] H. Hashimoto, M. Yagiura, and T. Ibaraki, An iterated local search algorithm for the time-dependent vehicle routing problem with time windows, *Working paper*.

[3] T. Ibaraki, S. Imahori, M. Kubo, T. Masuda, T. Uno, and M. Yagiura, Effective local search algorithms for routing and scheduling problems with general time-window constraints, *Transportation Science*, vol. 39, pp. 206–232, 2005.

[4] S. Ichoua, M. Gendreau, and J.Y. Potvin, Vehicle dispatching with time-dependent travel times, *European Journal of Operational Research*, vol. 144, pp. 379–396, 2003.

[5] E. Taillard, P. Badeau, M. Gendreau, F. Guertin, and J. Y. Potvin, A tabu search heuristic for the vehicle routing problem with soft time windows, *Transportation Science*, vol. 31, pp. 170–186, 1997.