AN OPTIMIZATION ALGORITHM FOR A GENERALIZED PICKUP AND DELIVERY PROBLEM WITH TIME WINDOWS

Jayoung Kang[1], M.S.,
Hee Jeong Zang[1], M.S.,
Jangha Kang[1], Ph.D.,
Sungsoo Park[1*], Professor
[1]KAIST, Department of Industrial Engineering, Korea Advanced Institute of Science and Technology, 373-1 Guseong-dong, Yuseong-gu, Daejeon, 305-701, Korea

ABSTRACT
The pickup and delivery problem with time windows (PDPTW) involves the construction of optimal routes which satisfy a set of transportation requests under capacity, time window, pairing, and precedence constraints. A transportation request is characterized by a pickup location and a delivery location. Each transportation request must be served by a vehicle. A vehicle may serve multiple transportation requests as long as other constraints are satisfied. We consider a generalized pickup and delivery problem with time windows (GPDPTW), in which a transportation request consists of a pickup location and many delivery locations. Here, a transportation request must be served by the same vehicle, too. In this paper, we propose a branch and price algorithm for the problem. An enumeration technique is used for the column generation problem. We tested the algorithm on randomly generated instances and computational results are reported. **KEY WORDS:** Transportation, Pickup and delivery problem with time windows, Branch-and-price algorithm

1. Introduction

In the pickup and delivery problem with time windows(PDPTW), an optimal set of routes has to be constructed to satisfy transportation requests. Each transportation request is specified by a pickup location, a delivery location, and load of items to be delivered. A route should end at its starting depot where vehicles are stationed. A vehicle is characterized by fixed cost, capacity and a depot. All routes are restricted by pairing, precedence, capacity, and time windows constraints. Pairing constraints ensure that a pickup location and a delivery location of a transportation request should be visited by one vehicle. To serve a transportation request, a vehicle collects items at a pickup location and delivers them to a delivery location without any transshipment at an intermediate location. Precedence constraints imply that a vehicle should visit the pickup location before the delivery location of a

* The corresponding author. E-mail: sspark@kaist.ac.kr , Tel: +82-42-869-3121

transportation request. Capacity constraints guarantee that load of items on a vehicle should be less than the vehicle capacity. Each location specifies a time window which is defined as a time interval between the earliest arrival time and the latest arrival time. Time windows constraints make sure that a service has to be given between the earliest arrival time and the latest arrival time. In this paper, we consider a generalized pickup and delivery problem with time windows(GPDPTW). In the GPDPTW, each transportation request may be specified by a pickup location and several delivery locations, and all the locations of a transportation request should be visited by one vehicle satisfying precedence, capacity, and time windows constraints because any transportation request can't be divided into several ones.

The GPDPTW is NP-hard by restriction, since the PDPTW is NP-hard and is a special case of the GPDPTW(Desrosiers, Dumas, Solomon, & Soumis, 1995). We can not find any algorithm for the GPDPTW. However, there exist some literatures of the PDPTW. The first optimization algorithm for the PDPTW was a branch-and-price algorithm presented by Dumas, Desrosiers, & Soumis(1991). It uses a dynamic programming algorithm to solve the subproblem. Savelsbergh, & Sol(1998) proposed a branch-and-price algorithm for the PDPTW using both a heuristic algorithm and a dynamic programming algorithm for the column generation problem. They applied a new branching scheme based on assignment rather than routing decisions. Savelsbergh & Sol(1995) presented definition and an integer programming formulation of the general pickup and delivery problem(GPDP) which considered several pickup and delivery locations of a transportation request. They also demonstrated necessity for researches on the GPDP which considers either one pickup location and several delivery locations or several pickup locations and one delivery location. We developed an optimization algorithm for the GPDPTW associated with one pickup and several delivery locations of a request and it was based on two algorithms proposed by Dumas, Desrosiers, & Soumis(1991) and Savelsbergh & Sol(1998).

This paper is organized as follows: Section 2 presents assumptions and a set partitioning model of the GPDPTW. Section 3 describes a branch-and-price algorithm for the GPDPTW. Section 4 shows the computational results. Finally, conclusions are given in section 5.

2. Model

We consider several vehicle types and assume that the number of available vehicles is given for each type. It can happen that a vehicle arrives at a location before the earliest arrival time. Then the vehicle can wait until the earliest arrival time, while it should not arrive after the latest arrival time. We assume that the service time of a location is zero since it is generally much less than the travel time. We suppose that travel time between two locations is independent of vehicle types. However, different travel time according to various vehicle types can be considered. The following notation is used:

N the set of transportation requests

M the set of vehicle types

m_k the number of available vehicles of type $k \in M$

Ω_k the set of all feasible routes for type $k \in M$

c_r^k the cost of a route $r \in \Omega_k$

$$\delta_{lr}^k = \begin{cases} 1 & \text{if a transportation request } l \text{ is served on a route } r \in \Omega_k \\ 0 & \text{o.w.} \end{cases}$$

A feasible route satisfies pairing, precedence, capacity, and time windows constraints. The decision variables are as follows:

$$x_r^k = \begin{cases} 1 & \text{if a route } r \in \Omega_k \text{ is used} \\ 0 & \text{o.w.} \end{cases}$$

Savelsbergh & Sol(1998) presented the following formulation for the PDPTW which can be used for the GPDPTW.

$$\text{Min} \quad \sum_{k \in M} \sum_{r \in \Omega_k} c_r^k x_r^k \tag{1}$$

$$\text{s.t.} \quad \sum_{k \in M} \sum_{r \in \Omega_k} \delta_{lr}^k x_r^k = 1 \qquad \text{for all } l \in N \tag{2}$$

$$\sum_{r \in \Omega_k} x_r^k \leq m_k \qquad \text{for all } k \in M \tag{3}$$

$$x_r^k \in \{0, 1\} \qquad \text{for all } k \in M,\ r \in \Omega_k \tag{4}$$

The objective is to minimize the total cost. The cost of a route can be calculated as the sum of fixed cost of the vehicle and travel costs. We assume that the travel cost between two locations is equal to the travel distance since the costs are primarily determined by the distance. Constraints (2) impose that each transportation request must be satisfied exactly once. Constraints (3) ensure that the number of used vehicles of each type is less than or equal to the number of available vehicles. Each column vector corresponds to a feasible route. Generally there are exponential numbers of columns. Thus, it is impractical to enumerate all possible columns. However, we can solve the LP relaxation efficiently by the column generation method and it generally gives a good bound.

3. Algorithm

We developed a branch-and-price algorithm for the GPDPTW. Since there are too many columns to enumerate, the LP relaxation of the problem is to be solved with a subset of all possible columns. If the minimum reduced cost is less than zero, the corresponding column is to be added to the restricted problem and then the problem is to be reoptimized. Let x be optimal to the restricted problem. Then the reduced cost of each route is nonnegative if and only if x is optimal to the original problem. Therefore, we repeat the procedure until no more columns with negative reduced costs are found. If an optimal solution is not integral, we need to explore a branch-and-bound tree. We generate columns at each branch-and-bound tree node.

The subproblem can be divided into several ones according to vehicle types. Each subproblem can be regarded as a constrained shortest path problem with time windows in which an origin and a destination are the depot. We solve the subproblem with an enumeration technique based on the dynamic programming algorithm proposed by Dumas, Desrosiers, & Soumis(1991). Preprocessing steps like the shrinking of the time windows and the elimination of the inadmissible arcs are performed before the enumeration starts. We use labels to enumerate columns corresponding to paths. The following notation is used:

P_i^g	the path g from the departure of a depot to location i
S_i^g	the set of locations visited on the path g
T_i^g	the time of service at location i on the path g
Z_i^g	the reduced cost at location i on the path g
$R(S_i^g)$	the set of locations which have to be visited
N_l^+	the set of pickup location of a transportation request $l \in N$, $\left\|N_l^+\right\| = 1$
N_l^-	the set of delivery locations of a transportation request $l \in N$
N^-	the set of all delivery locations
a_i	the earliest arrival time at location i
b_i	the latest arrival time at location i
c_{ij}	the travel distance from location i to j
F_k	the fixed cost of a vehicle type $k \in M$
u_l	the dual variable associated with constraint (2) for $l \in N$
v_k	the dual variable associated with constraint (3) for $k \in M$

$$c'_{ij} = \begin{cases} c_{ij} - u_l & \text{if } i \in N_l^+ \\ F_k + c_{ij} - v_k & \text{if } i \text{ is a depot of vehicle type } k \in M \\ c_{ij} & \text{o.w.} \end{cases}$$

A label $(S_i^g, R(S_i^g), T_i^g, Z_i^g)$ of the path P_i^g is considered, and it is sufficient to verify pairing, precedence, capacity, and time windows constraints. Assuming that location 0 is a depot, we start enumeration process with a label (∅, ∅, a_0, 0). Given a label $(S_i^g, R(S_i^g), T_i^g, Z_i^g)$ of the path P_i^g , an attempt to extend the path P_i^g to a path $P_j^{g'}$ can be made if there is an arc (*i*, *j*). A new label $(S_j^{g'}, R(S_j^{g'}), T_j^{g'}, Z_j^{g'})$ associated with the path $P_j^{g'}$ can be calculated as follows:

$$S_j^{g'} = S_i^g \cup \{j\}$$

$$R(S_j^{g'}) = \begin{cases} R(S_i^g) \cup \{h \mid h \in N_l^-\} & \text{if } l \in N \text{ and } j \in N_l^+ \\ R(S_i^g) \setminus \{j\} & \text{if } j \in N^- \text{ and } j \in R(S_i^g) \\ \text{infeasible} & \text{o.w.} \end{cases}$$

$$T_j^{g'} = \begin{cases} \max\{a_j,\ T_i^g + t_{ij}\} & \text{if } T_j^{g'} \le b_j \\ \text{infeasible} & \text{o.w.} \end{cases}$$

$$Z_j^{g'} = Z_i^g + c_{ij}^{'}$$

Labels denoting infeasible paths should be eliminated. We can also eliminate labels that will be inadmissible in the future (Dumas, Desrosiers, & Soumis, 1991). If the last node of $S_j^{g'}$ is a depot, the path $P_j^{g'}$ is a completed path, that is, a route. If the route is feasible and $Z_j^{g'} < 0$, the corresponding column can be added into the restricted problem.

If an optimal solution *x* to the LP relaxation of the problem is fractional, we solve the mixed integer programming problem with a given set of columns using CPLEX callable mixed integer library. The integral solution can provide an upper bound. Supposing that *x* is fractional, and $y_{ij} = \sum_{k \in M} \sum_{r \in \Omega_k} \delta_{ir}^k \delta_{jr}^k x_r^k$, there must be two requests $i, j \in N$ satisfying $0 < y_{ij} < 1$. Then we can divide into two subsets characterized by $y_{ij} = 0$ and $y_{ij} = 1$ (Sol & Savelsbergh). We can generate columns at any branch-and-bound tree node if we use an adjusted enumeration method which is similar to the previous one.

4. Computational experiments

We used CPLEX 8.1 callable library to solve LP and the branch-and-price algorithm was tested on a Pentium PC (2.4GHz). Three types of problems were randomly generated. The problem of type A is the PDPTW. The problem of type B is the GPDPTW. The problem of type C involves transportation requests with more than two delivery locations. We tested the algorithm on ten instances for each problem set. The problem set A20 consists of ten instances of type A and each instance considers twenty service locations. The characteristics of other problem sets can be specified in the same way. Computational results are given in the following table.

Table 1. The test result

	A20	A30	A40	A50	B20	B30	B40	B50	C30	C40
A_GAP	0.54	1.12	0.29	0.95	1.11	1.01	2.14	0.23	0.84	1.2
A_Cols	35.2	135.4	217.8	1769.4	22.4	63.6	95.7	51.6	29.8	34.9
A_Time	1.98	16.27	182.05	1365.78	0.75	34.72	307.81	178.27	8.08	7.65

Let Z_{LP} and Z_{IP} be the optimal values of the LP relaxation problem and the original problem respectively. Then the GAP(%) is calculated as $(Z_{IP} - Z_{LP}) / Z_{IP} \times 100$ and A_GAP(%) is the average GAP of the problem set. A_Cols(unit) represents the average number of generated columns. A_Time(sec) is the average computational time. Seventy-seven instances are solved within a minute and all instances are solved within an hour. The average computational times of type A, B, and C are 391.52, 130.38, and 7.865 individually. As the number of delivery locations of a request increases, the computational time generally decreases. It implies that the large number of delivery locations of a request may decrease the size of solution space.

5. Conclusions

In this paper, we developed a branch-and-price algorithm for the GPDPTW. We used the enumeration technique to generate columns with preprocessing and some elimination rules. We tested the algorithm on randomly generated problems and the results showed that this algorithm can provide optimal solutions quickly when the number of locations associated with transportation requests is large and the problems are restrictive on capacity and time windows. Since the PDPTW is a special case of the GPDPTW, it can be solved by this algorithm, too. Although an optimization algorithm for the GPDPTW has been suggested, there still remain other research topics about the GPDPTW. For instance, we did not consider some restrictions on labor time or route duration in practical situations. So, it can be a good research work to develop an algorithm considering such restrictions.

References

Desrosiers, J., Dumas, Y., Solomon, M.M., & Soumis, F.(1995). Time constrained routing and scheduling. In: M.O. Ball, T.L. Magnanti, C.L. Monma, & G.L. Nemhauser (eds.), *Network Routing, Handbooks in Operations Research and Management Science, vol*. 8 (pp. 35-139). INFORMS: Elsevier Science.

Dumas, Y., Desrosiers, J. & Soumis, F.(1991). The pickup and delivery problem with time windows. *European Journal of Operational Research*, 54, 7-22.

Nanry, W. P. & Barnes, J.W.(2000). Solving the pickup and delivery problem with time windows using reactive tabu search. *Transportation Research* Part B 34, 107-121.

Savelsbergh, M.W.P. & Sol, M.(1995). The general pickup and delivery problem. *Transportation Science*, 29(1), 17-29.

Savelsbergh, M.W.P. & Sol, M.(1998). DRIVE: Dynamic routing of independent vehicles. *Operations Research*, 46(4), 474-490.

Sol, M. & Savelsbergh, M.W.P.. A branch-and-price algorithm for the pickup and delivery problem with time windows. *http://www.isye.gatech.edu/ research/files/lec 9406.pdf.*