# AN ALGORITHM FOR
# THE SINGLE PICKUP AND DELIVERY PROBLEM WITH TIME WINDOWS

Jayoung Kang
Department of Industrial Engineering, KAIST, kangdding@kaist.ac.kr

Hee Jeong Zang
Department of Industrial Engineering, KAIST, eyest@kaist.ac.kr

Jangha Kang
Department of Industrial Engineering, KAIST, Jangha.kang@kaist.ac.kr

Sungsoo Park
Department of Industrial Engineering, KAIST, sspark@kaist.ac.kr

## ABSTRACT

The pickup and delivery problem with time windows involves the construction of optimal routes which satisfy a set of transportation requests under pairing, precedence, time window, vehicle capacity, availability, location capacity, distance, duration and extra worker constraints. We consider three types of the objectives which are involved in minimizing the total travel cost, the total travel distance and the total travel duration. A transportation request is characterized by a pickup location, a delivery location and items to be delivered. Each transportation request must be served by a vehicle. A vehicle may serve multiple transportation requests as long as other constraints are satisfied. 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.

**KEYWORDS:** Vehicle routing problem, Pickup and delivery problem with time windows, Branch-and-price algorithm

## INTRODUCTION

In the pickup and delivery problem with time windows(PDPTW), an optimal set of routes has to be constructed to satisfy transportation requests. A transportation request is characterized by a pickup location, a delivery location and items to be delivered in the single PDPTW. A transportation request is specified by several pickup and delivery locations in the general PDPTW(see Savelsbergh et al. (1995)). We consider the single PDPTW in this paper and the PDPTW refers to the single PDPTW from now on. A route should end at its starting depot where vehicles are stationed. We consider several types of vehicles and each vehicle is characterized by cost factors, capacity and a depot. All routes are restricted by pairing, precedence, time window, vehicle capacity, availability, location capacity, distance, duration and extra worker constraints. Pairing constraints ensure that a pickup location and a delivery location of each transportation request should be visited by one vehicle. To satisfy 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 transportation request. Each location specifies a time window which is defined as a time interval between the earliest arrival time and the latest arrival time. Time window constraints make sure that a service at a location has to be given between the earliest arrival time and the latest arrival time of the location. Vehicle capacity constraints guarantee that load of items on a vehicle should be less than or equal to the vehicle capacity. We assume that the number of available vehicles for each vehicle type is limited. So availability constraints ensure that the number of used vehicles is less than or equal to the number of available vehicles for each vehicle type. We assume that location capacity for each location is given since a big vehicle cannot enter a small parking lot. If the capacity of a vehicle is greater than the given location capacity of a location, the vehicle cannot serve at the location since we consider location constraints. We consider distance constraints and duration constraints since there is the upper limit of working hours in some fields of industry. Distance constraints and duration constraints guarantee that the travel distance and duration of a route should be less than or equal to the given upper limits respectively. If an item is so heavy that a driver cannot carry it alone, extra workers are needed to deliver the item. Therefore, we consider extra worker constraints which make sure that the number of necessary extra workers of a route is greater than or equal to the maximum number of extra workers of items to be delivered by the route. In this paper, three objectives are considered. The first objective is to minimize the total travel cost which is the sum of the costs of all routes. The cost of a route consists of the fixed cost, the distance-based cost and the time-based cost. The fixed cost is determined by the vehicle type. The distance-based cost is directly proportional to the travel distance. The time-based cost are influenced by labor costs and working hours. The second objective is to minimize the total travel distance. The third objective is to minimize the total travel duration. The ending time minus the starting time of a route equals the travel duration of a route.

The PDPTW is NP-hard by restriction, since the vehicle routing problem(VRP) is NP-hard and it is a special case of the PDPTW(see Desrosiers et al.(1995)). There exist some literatures of the PDPTW. Dumas et al.(1991) presented an optimization algorithm for the

PDPTW which was to construct optimal routes satisfying a set of transportation requests under pairing, precedence, time window, vehicle capacity and availability constraints. They considered the minimization of the total travel cost as an objective and the total travel cost was determined by the load of vehicles, the travel distance and the number of necessary vehicles. The presented algorithm is a branch-and-price algorithm, and uses a dynamic programming algorithm to solve the subproblem. Sol et al. proposed a branch-and-price algorithm for the PDPTW which was same as the problem considered by Dumas et al.(1991) except the objective. The primary objective considered by Sol et al. was the minimization of the number of necessary vehicles to satisfy all transportation requests. The second objective was to minimize the total travel distance. The branch-and-price algorithm uses a heuristic algorithm and the dynamic programming algorithm for the column generation problem. They applied a new branching scheme based on assignment rather than routing decisions. In those papers, location capacity, distance, duration and extra worker constraints were not considered. In addition, the dynamic algorithm they presented fixed the starting time of a route to the earliest arrival time of a depot. It is sufficient to provide an optimal solution of the existing PDPTW because the objective they considered is not affected by the travel time. However, the travel cost is influenced by the travel time in this paper. Therefore we developed another algorithm which does not fix the starting time of a route and is compatible with all constraints.

## 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 should 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 can be included into the travel time. Different travel times are considered according to the vehicle types.

If we can find all possible routes, the PDPTW to construct an optimal route can be converted to the problem which is to decide whether we use a route or not for each feasible route. The converted problem is called as the master problem. The following notation is used to model the master problem:

| | |
|---|---|
| $N$ | the set of transportation requests |
| $M$ | the set of vehicle types |
| $m_k$ | the number of available vehicles of type $k \in M$ |
| $\Omega_k$ | the set of all feasible routes for type $k \in M$ |

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

$$c_{r}^{k} = \begin{cases} \text{the travel cost of a route } r \in \Omega_k, & \text{if the objective is to minimize the total travel cost} \\ \text{the travel distance of a route } r \in \Omega_k, & \text{if the objective is to minimize the total travel distance} \\ \text{the travel duration of a route } r \in \Omega_k, & \text{if the objective is to minimize the total travel duration} \end{cases}$$

A feasible route satisfies pairing, precedence, time window, vehicle capacity, location capacity, distance, duration and extra worker constraints. The decision variables are as follows:

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

The following formulation is for the master problem:

[MP]

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

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

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

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

The objective function is represented by (1). Constraints (2) impose that each transportation request must be satisfied exactly once. Constraints (3) represents availability constraints. A feasible route $r \in \Omega_k$ corresponds to a column vector $(\delta_{1r}^k, \delta_{2r}^k, \delta_{3r}^k, ..., \delta_{|N|r}^k)'$. Since there are generally exponential numbers of columns, it is impractical to enumerate all possible columns. However, we can solve the master problem without enumerating all feasible columns by the column generation method. And also, the linear programming(LP) relaxation of the master problem usually gives a good bound.

**ALGORITHM**

We developed a branch-and-price algorithm for the PDPTW in this paper. Since there are too many columns to enumerate, the LP relaxation of the problem should be solved with a subset of all possible columns. The master problem with the subset of all possible columns is the restricted master problem. Although it is not trivial to construct an initial subset of columns, we can initialize the restricted master problem by the two-phase method(see Sol et al.). We optimize the LP relaxation of the restricted master problem instead of the LP relaxation of the master problem. The subproblem is to find the column with the minimum reduced cost. If the minimum reduced cost is less than zero, there is at least one column with negative reduced cost. We can add the column with negative reduced cost to the LP relaxation of the restricted master problem and reoptimize it. Otherwise, all possible columns have nonnegative reduced costs and a current optimal solution of the LP relaxation of the restricted master problem can be an optimal solution of the LP relaxation of the master problem. Therefore, we repeat the procedure until no more columns with negative reduced costs are found. If an optimal solution of the master problem 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. We can construct a graph for the divided problem where each location is a node and each path between two locations is an arc. If we duplicate the depot as the origin depot and the destination depot, the divided subproblem can be regarded as a constrained shortest path problem with time windows. We solve the problem with an enumeration technique based on the dynamic programming algorithm proposed by Dumas et al.(1991). Preprocessing steps such as the shrinking of the time windows and the elimination of the inadmissible arcs are performed before the enumeration starts. Location capacity constraints can be ensured by eliminating the inadmissible arcs. The following notation is used:

| | |
|---|---|
| $0^{+}$ | the origin depot |
| $0^{-}$ | the destination depot |
| $P_i^q$ | the path $q$ from the departure of a depot to node $i$ |
| $S_i^q$ | the set of nodes visited on the path $q$ |
| $R(S_i^q)$ | the set of nodes which have to be visited by the path $q$ from the departure of node $i$ to the destination depot |
| $T_i^q$ | the time of service at node $i$ on the path $q$ |
| $X_i^q$ | the departure time of the origin depot on the path $q$ |
| $I_i^q$ | 0 if the departure time of the origin depot on the path $q$ can be postponed,<br>1 o.w. |
| $Z_i^q$ | the travel distance from the origin depot to node $i$ on the path $q$ |
| $p_l$ | the pickup node of a transportation request $l \in N$ |
| $d_l$ | the delivery node of a transportation request $l \in N$ |
| $t_{ij}^k$ | the travel time from node $i$ to $j$ by a vehicle type $k \in M$ |

| | |
|---|---|
| $a_i^k$ | the earliest arrival time at node *i* by a vehicle type $k \in M$ |
| $b_i^k$ | the latest arrival time at node *i* by a vehicle type $k \in M$ |
| $d_{ij}$ | the travel distance from node *i* to *j* |
| $U_d$ | the upper limit for the travel distance |
| $U_t$ | the upper limit for the travel duration |

Each path $P_i^q$ is related with label $(S_i^q, R(S_i^q), T_i^q, X_i^q, I_i^q, Z_i^q)$. A label contains all information to verify satisfaction of constraints.

**Enumeration technique**

Step 1. Add the path $P_{0^+}^0$ with label $(\{0^+\}, \varnothing, a_{0^+}, a_{0^+}, 0, 0)$ to an empty path list. Initialize the minimum reduced cost as zero. Initialize the minimum reduced cost path as empty.

Step 2. If the path list is empty, stop. Otherwise, choose a path from the path list.

Step 3. Given a path $P_i^q$ with label $(S_i^q, R(S_i^q), T_i^q, X_i^q, I_i^q, Z_i^q)$, go to step 4 if the last visited node *i* of the path $P_i^q$ is not $0^-$. If the path does not satisfy pairing, or duration constraints, discard the path $P_i^q$ and go to step 2. If the reduced cost of the path $P_i^q$ is less than the current minimum reduced cost, update the minimum reduced cost and the minimum reduced cost path. And then, go to step 2.

Step 4. Given a path $P_i^q$ with label $(S_i^q, R(S_i^q), T_i^q, X_i^q, I_i^q, Z_i^q)$, try to extend the path $P_i^q$ to node *j* which is not an element of the set $S_i^q$ for all existing arc (*i*, *j*). If there is no arc to be added, discard the path $P_i^q$.

When we try to extend the path $P_i^q$ to node *j* at step 4, the new label $(S_j^{q'}, R(S_j^{q'}), T_j^{q'}, X_j^{q'}, I_j^{q'}, Z_j^{q'})$ of the extended path $P_j^{q'}$ is calculated as follows:

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

$$Z_j^{q'} = \begin{cases} Z_i^q + d_{ij} & \text{if } Z_i^q + d_{ij} \le U_d \\ \text{infeasible} & \text{o.w.} \end{cases}$$

$$R(S_j^{q'}) = \begin{cases} R(S_i^q) \cup \{d_l\} & \text{if } l \in N \text{ and } j = p_l \\ R(S_i^q) \setminus \{j\} & \text{if } j \in R(S_i^q) \\ \text{infeasible} & \text{o.w.} \end{cases}$$

Case 1. $I_i^q = 0, T_i^q + t_{ij}^k < a_j^k$

$\Rightarrow$ $T_j^{q'} = a_j^k$. Calculate back to $X_j^{q'}$ and $I_j^{q'}$ from $T_j^{q'}$.

Case 2. $I_i^q = 1, T_i^q + t_{ij}^k < a_j^k$

$\Rightarrow$ $T_j^{q'} = a_j^k$, $X_j^{q'} = X_i^q$, $I_j^{q'} = I_i^q$.

Case 3. $a_j^k \le T_i^q + t_{ij}^k \le b_j^k$

$\Rightarrow$ $T_j^{q'} = T_i^q + t_{ij}^k$, $X_j^{q'} = X_i^q$, $I_j^{q'} = I_i^q$.

Case 4. $T_i^q + t_{ij}^k > b_j^k$

$\Rightarrow$ The path $P_j^{q'}$ is infeasible since it violates time window constraints.

We can easily verify vehicle capacity constraints using the set $S_j^{q'}$. If node *j* is a pickup node of a transportation request, the delivery node of the transportation request is added to the set $R(S_j^{q'})$. If node *j* is a delivery location which is an element of the set $R(S_i^q)$, it is deleted from the set. If node *j* is a delivery location which does not belong to the set $R(S_i^q)$, the path $P_j^{q'}$ violates precedence constraints and the above calculating process should be stopped. And at step 3, if node j of the path $P_j^{q'}$ is $0^-$ and the set $R(S_j^{q'})$ is not empty**,** the path $P_j^{q'}$ violates pairing constraints. So, we can check pairing and precedence constraints by the set $R(S_j^{q'})$. While calculating $Z_j^{q'}$ and $T_j^{q'}$, we can certify distance and time window constraints. If the path $P_j^{q'}$ satisfies $I_j^{q'} = 1$ and $T_j^{q'} - X_j^{q'} > U_t$, it violates duration constraints. So, we can discard it.

In case 1, $X_j^{q'}$ is reckoned backward from $T_j^{q'}$ since the vehicle arrive at node *j* before the earliest arrival time and the starting time of the path can be postponed. The following procedure is for calculating $X_j^{q'}$ and $I_j^{q'}$ assuming that node *i* is visited just before node *j* and $t_i$ is the service time at node *i* on the path $P_j^{q'}$.

$$T_j^{q'} = a_j^k,\ I_j^{q'} = 0.$$

$$t_i = \begin{cases} T_j^{q'} - t_{ij}^k & \text{if } a_i^k \le T_j^{q'} - t_{ij}^k \le b_i^k \\ b_i^k & \text{if } T_j^{q'} - t_{ij}^k > b_i^k \end{cases}$$

$$I_j^{q'} = 1 \qquad \text{if } t_i = b_i^k$$

Replace node *j* and *i* as node *i* and the node which is visited just before node *i* respectively**.** And then, repeat the above procedure until $X_j^{q'}$ can be obtained.

$T_j^{q'} - t_{ij}^k < a_i^k$ cannot be happen. Infeasible paths should be eliminated. And also, paths which will be inadmissible in the future can be eliminated while extending it(see Dumas et al.(1991)).

If an optimal solution $x$ to the LP relaxation of the master problem is fractional, we solve the restricted master problem using CPLEX callable mixed integer library. The integral solution can provide an upper bound. And then, we explore the branch-and-bound tree. 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 $i$ and $j \in N$ satisfying $0 < y_{ij} < 1$. Then we can divide feasible region into two subsets characterized by $y_{ij} = 0$ and $y_{ij} = 1$ (see Sol et al.). 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.

## 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 tested. The objective of the type A is to minimize the total travel cost. The objective of type B is to minimize the total travel distance. The objective of type C is to minimize the total travel duration. First, we randomly generated thirty instances of type A. We made instances of type B and C by replacing the objectives of those instances. So, we tested the algorithm on ninety instances. The problem set A20 consists of ten instances of type A and each instance includes twenty service locations. The characteristics of other problem sets can be specified in the same way. That is, the number of service locations we considered varies from twenty to forty. Computational results are given in the following table.

**Table 1. The test result**

| | $Z_{LP}$ | $Z_{IP}$ | A_GAP | A_B&B | A_COLS | A_TIME |
|---|---|---|---|---|---|---|
| A20 | 5733.328 | 5746.387 | 0.152 | 2.0 | 49.1 | 1.565 |
| A30 | 6775.731 | 6940.265 | 2.114 | 78.8 | 3097.6 | 375.512 |
| A40 | 12227.410 | 12317.520 | 1.580 | 54.6 | 3145.7 | 1107.938 |
| B20 | 616.234 | 617.600 | 0.249 | 1.6 | 40.3 | 1.342 |
| B30 | 866.889 | 873.665 | 0.758 | 7.0 | 276.6 | 63.406 |
| B40 | 1259.957 | 1263.640 | 0.298 | 6.8 | 425.4 | 310.390 |
| C20 | 718.660 | 721.609 | 0.413 | 1.8 | 48.0 | 1.490 |
| C30 | 1005.972 | 1013.670 | 0.751 | 7.6 | 320.6 | 72.041 |
| C40 | 1466.140 | 1471.598 | 0.381 | 6.6 | 382.2 | 275.243 |

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_B&B is the average number of generated nodes in the branch-and-bound tree. A_COLS(unit) represents the average number of generated columns. A_TIME(sec) is the average computational time to solve the original problem optimally. Most GAP values were less than five percent. Forty-five instances were solved without branching and the maximum number of branch-and-bound nodes was six hundred five. A_TIME increases as the number of transportation requests increases. The number of generated columns in a problem which minimizes the total travel cost is usually greater than the number of generated columns in the other problems. Therefore, instances of type A usually took more time to be solved optimally than instances of other types.

## CONCLUSIONS

In this paper, we developed a branch-and-price algorithm for the PDPTW which considered several realistic constraints and objectives. We used the enumeration technique to generate columns with preprocessing and some elimination rules. We can consider the objectives and constraints which are related with the travel time since postponement of the starting time at a depot is possible in our algorithm. We tested the algorithm on randomly generated problems and the results showed that this algorithm can provide an optimal solution for the PDPTW in a proper time.

## REFERENCES


Barnhart, C., E.L. Johnson, G.L. Nemauser, M.W.P. Savelsbergh and P.H. Vance, “Branch-and-Price: Column Generation for Solving Huge Integer Programs,” *Operations Research*, 46, 1998, pp.316-329.

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

Dumas, Y., J. Desrosiers, and F. Soumis, “The Pickup and Delivery Problem with Time Windows,” *European Journal of Operational Research*, 54, 1991, pp.7-22.

Nanry, W. P. and J.W. Barnes, “Solving the Pickup and Delivery Problem with Time Windows Using Reactive Tabu Search,” *Transportation Research*, Part B 34, 2000, pp.107-121.

Nemhauser, G.L. and L. Wolsey, “Integer and Combinatorial Optimization,” *Wiley*, 1988.

Savelsbergh, M.W.P. and M. Sol, “The General Pickup and Delivery Problem,” *Transportation Science*, 29, 1995, pp.17-29.

Savelsbergh, M.W.P. and M. Sol, “DRIVE: Dynamic Routing of Independent Vehicles,” *Operations Research*, 46, 1998, pp.474-490.
Sol, M. M.W.P. Savelsbergh, “A Branch-and-Price Algorithm for the Pickup and Delivery Problem with Time Windows,” *http://www.isye.gatech.edu/ research/files/lec 9406.pdf.*