Exact Solution for Vehicle Routing Problem with Soft Time Windows and Dynamic Travel Time

Ali Gul QURESHI
Program Specific Senior Lecturer
Graduate School of Engineering
Kyoto University
C-1, Kyotodaigaku, Katsura Campus,
Nishikyo-ku, Kyoto
615-8540, Japan
Fax: +81-75-950-3800
E-mail: aligul@kiban.kuciv.kyoto-u.ac.jp

Eiichi TANIGUCHI
Professor
Graduate School of Engineering
Kyoto University
C-1, Kyotodaigaku, Katsura Campus,
Nishikyo-ku, Kyoto
615-8540, Japan
Fax: +81-75-950-3800
E-mail: taniguchi@kiban.kuciv.kyoto-u.ac.jp

Tadashi YAMADA
Associate Professor
Graduate School of Engineering
Kyoto University
C-1, Kyotodaigaku, Katsura Campus,
Nishikyo-ku, Kyoto
615-8540, Japan
Fax: +81-75-950-3800
E-mail: t.yamada@kiban.kuciv.kyoto-u.ac.jp

Abstract: The traffic conditions in urban areas change with time due to varying congestion levels and incidents resulting in unexpected varying travel time on the infrastructure links. These conditions lead to the formulation of the Dynamic Vehicle Routing and scheduling Problem with Soft Time Windows (D-VRPSTW) with dynamic travel time in the city logistics field. This paper presents a column generation-based exact solution approach for the D-VRPSTW. The performance evaluation on a test instance under different dynamic travel time events, shows that the D-VRPSTW solutions based on the updated travel times result in significant cost savings as compared to static version.

Key Words: City logistics, Dynamic vehicle routing, Column generation, Dynamic travel time

1. INTRODUCTION

Industrial growth and expanding employment opportunities have led to the urban-oriented economic development in many countries. Demand of transportation, both in terms of passengers as well as for freight is also increasing in and around these big urban conurbations. Door to door service industries such as home appliance or utilities repair services, are also considered in the broader definition of the urban logistics. Transportation cost shares a considerable proportion in the prices of such services and many other products. Almost all of the intra-urban freight movement (both pickup and deliveries) is carried out by trucks and/or vans, for example in Tokyo, Japan, the share of road-based transport is about 99.4% of the total intra-urban freight movement (Bureau of Industrial and Labor Affairs in Tokyo Metropolitan Government, 2009). Traffic congestion, noise, vibrations, generation of NOx, SPM, CO\textsubscript{2} and other environmental problems, crashes, loading and unloading on the street side are typical problems caused by the road-based freight transport in urban areas. City logistics deals with the optimization of logistics and transport activities within an urban area by considering costs and benefits for both public and private sectors, not only in terms of delivery cost but in terms of environment, traffic congestion and energy use; while still being
within the framework of market economy (Thompson and Taniguchi, 2001).

To be competitive in the logistics business, the logistics services have also improved for example the introduction of time windows which promises delivery/pickup within specified time slots. Such innovations in logistics services often increase the cost and reduce the load factor. Logistics firms try to cover these increased costs by improving their operational efficiency. Route optimization is one of such efficiency improvement measures, which not only serves the logistics firms, it also advances the objectives of other city logistics stakeholders (for example, city administration and residents) as it minimizes the number of delivery vehicles and their traveled distance thus help reducing congestion, emissions and safety problems as well. The Vehicle Routing and scheduling Problem with Time Windows (VRPTW) is a commonly adopted route optimization technique, which consists of finding a set of minimum cost routes (for delivery vehicles) to cover the demands (weights to be picked up or delivered) of all customers within their specified time windows \([a_i, b_i]\). In the soft time windows case i.e. the Vehicle Routing and scheduling Problem with Soft Time Windows (VRPSTW), if a vehicle arrives earlier it has to wait (without associated cost) until the start of time windows \(a_i\); whereas a late arrival after \(b_i\) is allowed at some penalty (for example see, Taillard et al., 1997; Gendreau et al., 1999). Another version of full soft time windows has also been used in literature that penalizes both late as well as early arrival of a vehicle (such as in Qureshi et al., 2010a; Qureshi and Hanaoka, 2005; Yamada et al. 2004; Taniguchi and Heijden 2000); however, in this research we will use the former definition of the soft time windows (i.e. same as in Taillard et al., 1997; Gendreau et al., 1999; Qureshi et al., 2010b).

The classical VRPTW is defined for static input values such as fixed customer locations and static travel time. However, the traffic conditions in urban areas change with time due to varying congestion levels and incidents resulting in varying travel time on the infrastructure links. With the introduction of the Intelligent Transportation Systems (ITS) such as the Vehicle Information and Communication System (VICS) in Japan, it is possible to collect, deliver and store such dynamic travel times on a link. As far as the logistics is concerned, changes in the travel time may affect the distribution or pick-up routes of the delivery vehicles resulting in unexpected long delays if the routing is fixed and based on a static value of travel time (such as the average travel time). The case of everyday congestion that can be predicted to a degree of certainty is handled in the stochastic versions of the VRPTW models. The dynamic case deals with situations which result in unexpected reduction/increase in travel time or travel cost such as incidents, dynamic pricing of the network, environmental pricing of the network (similar to dynamic pricing but with the intent of environmental and congestion control rather than network utilization) as well as the events of natural disasters. These traffic conditions in urban areas result in the Dynamic Vehicle Routing and scheduling Problem with Soft Time Windows (D-VRPSTW) in the field of city logistics. There exists very few heuristics approaches to solve this problem in the literature (see §2 for details); but these heuristics techniques do not guarantee to identify the exact solution or even state the gap between the exact and the heuristically optimized solution.

This paper presents a column generation-based exact solution approach for the D-VRPSTW that can be used in the dynamic traffic information environment. The travel time is updated whenever a vehicle reaches a customer; based on the updated travel time, the routes (for all vehicles) are re-optimized for the remaining customers. The rest of the paper is structured as follows: section 2 provides a literature review of related research while the section 3 formulates the VRPSTW and describes its Dantzig-Wolfe decomposition, which is the foundation of the column generation-based algorithms. Section 4 defines the methodology
used to handle the dynamic travel time within the column generation framework. The test instance and the scenarios of the dynamic travel time are given in section 5. Section 6 describes the implementation and discusses the results obtained on the test instances comparing the static and the dynamic versions of the VRPSTW. Finally, section 7 draws some conclusions and gives some future research prospects.

2. LITERATURE REVIEW

The column generation-based algorithms have been very popular in the exact optimization field for the Vehicle Routing and scheduling Problem with Hard Time Windows (VRPHTW), which does not allow any violations of the time windows (for example, see Desrochers et al., 1992; Kohl et al., 1997; Irnich and Villeneuve, 2003; Feillet et al., 2004; Chabrier, 2006). These algorithms have not only improved the size of the problems solved enormously, they also have significantly reduced the computational efforts required for the hard time windows variant. In the field of the VRPSTW research also, many researchers have worked with the set partitioning linear optimization; however, most of the research is in heuristics domain. For example, Calvete et al. (2007) exploited goal programming to enumerate all feasible routes and then used a set partitioning problem to solve the VRP with soft time windows with heterogeneous fleet and multi objectives. In a similar approach, Fagerholt (2001) solved a ship-scheduling problem with soft time windows. He used the Traveling Salesman Problem with Capacity, Hard Time Windows and Precedence Constraint (TSP-CHTWPC) to enumerate all feasible routes and then optimizes their schedule using soft time windows, before using a set partitioning problem. Instead of pre-generated columns, Qureshi et al. (2010a) developed a heuristics approach generating columns in each iteration utilizing the useful dual information obtained in the solution of the set partitioning formulation (i.e. the master problem).

A vast literature is available for the typical heuristics solutions of the VRPSTW as well. For example, Balakrishnan (1993) described three simple sequential insertion heuristics for the VRPSTW based on the nearest neighbor, Clarke-Wright savings and space-time rules. Hashimoto et al. (2006) used local search and dynamic programming in a route first schedule second algorithm to solve the VRPTW with soft time windows and variable travel time costs. First, routes are optimized using a local search algorithm and then the optimum service start time for each customer is found as a subproblem. Duin et al. (2007) used the VRPSTW model and solved it with a tabu search for each freight carrier in a framework of hybrid freight market, where a fraction of demands is pre-allotted and another fraction is available in a real time auctioning system. Genetic algorithms (GA) are often employed in solving complex and close to real life VRPSTW instances in city logistics; for example Yamada et al. (2001) combined the logistics terminal location, Cooperative Delivery System (CDS) and VRPSTW in a single framework. The combined model was solved using a GA heuristics and the results were compared to a base case that did not use the CDS. Qureshi and Hanaoka (2005) studied a CDS using GA solutions of the associated VRPSTW, where a truck assigning module assigns consolidated routes back to trucks of participating companies. Utilizing the VICS (Vehicle Identification and Communication System) data and the data from 66 days operation of probe pickup/delivery trucks, Ando and Taniguchi (2007) have applied the VRPSTW-P and its GA solution to an actual delivery system in Osaka, Japan. An excellent review of the heuristic methods applied to the VRPHTW and the VRPSTW can be found in Braysy and Gendreau (2005a; 2005b) and in Taniguchi et al. (2001).
While bulk of the research targeting the soft time windows is heuristics based, the exact solution research incorporating soft time windows has been focused on the schedule optimization of a given fixed path considering linear (Sexton and Bodin, 1985) and/or generalized convex penalty functions (Dumas et al., 1990). Recently, Tagmouti et al. (2007) have presented an arc routing problem with soft time windows, where vehicles are not allowed to wait along their routes. In their column generation scheme, they have used a modified labeling algorithm for the Shortest Path Problem with Time Windows and Time Costs (SPPTWTC) subproblem earlier given by Ioachim et al. (1998). The vehicle arrival pattern has been represented by a continuous variable that resulted in very high computation times and limited the maximum size of problem solved to 40 customers. Recently, Qureshi et al. (2009) presented a column generation based exact solution approach for the soft time windows that considers late arrival penalties, and efficiently solved instances up to 50 customers under various soft time windows settings.

However, all the above-mentioned approaches are based on the static information and lack the dynamic nature of the real life logistics operations. This research extends our earlier work on static version of the VRPSTW (i.e. Qureshi et al. (2009)) to develop a column generation based exact solution technique for the D-VRPSTW. A routing system, in which complete or a part of input information (such as number and location of customers or travel time on arcs) is not available to the decision maker at the start but it is revealed during the scheduling horizon (day of operation), and in which the decision maker reacts to this new information by evoking some sort of re-optimization mechanism is defined as the D-VRPSTW (Psaraftis, 1995). There are mainly two sources of dynamism, viz. the dynamic customers case, in which new service request (new customer) are called-in during the day of operation, and the dynamic travel time case that deals with the unexpected changes of the travel time on the network links. There exists an abundance of research on the dynamic customers case of the D-VRPTW (for example see, Larsen, 2001; Chen and Xu, 2006; Branchini et al., 2009); whereas the dynamic travel time case of the D-VRPTW has received a limited attention. In fact only two references could be found dealing with the dynamic travel times, viz. Taniguchi and Shimamoto (2004), and Flieschmann et al. (2004); both considering the Dynamic-Vehicle Routing and scheduling Problem with Soft Time Windows (D-VRPSTW). Taniguchi and Shimamoto (2004) have used a macro-simulation scheme to generate the dynamic travel time data for a theoretical test network, whereas, Flieschmann et al. (2004) have used the data from an ITS implemented in Berlin, Germany, named as LISB, which provides the travel time data on links for every 5 minutes slot. However, both of these research works have adopted heuristics approaches to solve the D-VRPSTW. The heuristic techniques are sometimes faster and easily implemented than exact solutions, yet they do not guarantee to identify the exact solution or state how close to the exact solution a particular feasible solution is (Thompson and Van Duin, 2003). Therefore, this research proposes a column generation based exact solution approach for the D-VRPSTW with dynamic travel times to fill the existing research gap. The exact approach can be used for small to medium instances (up to 50 customers) as well as for the evaluation and calibration of the heuristics approaches.

3. FORMULATION OF STATIC VRPSTW AND ITS DANTZIG-WOLFE DECOMPOSITION

3.1 The VRPSTW Formulation

The classical VRPSTW, which is static in its nature, is defined on a directed graph $G = (V, A)$. The vertex set $V$ includes the depot vertex 0 and a fixed set of customers $C = \{1, 2, \ldots, n\}$. 
The arc set $A$ consists of all feasible arcs $(i, j)$, $i, j \in V$. With every vertex of $V$ there is an associated demand $d_i$, with $d_0 = 0$, and a time window $[a_i, b_i]$ representing the earliest and the latest possible service start times. A set of identical vehicles (represented by $K$) each with a capacity $q$ stationed at the depot, is available to service customers’ demands. Both cost $c_{ij}$ as well as time $t_{ij}$ are associated with each arc $(i, j) \in A$, which remain same during the whole scheduling horizon. The time $t_{ij}$ includes the travel time on arc $(i, j)$ and the service time at vertex $i$, and a fixed vehicle utilization cost is added to all outgoing arcs from the depot, i.e. in $c_{0j}, j \in C$. However, based on a routing decision, the modified arc costs $c'_{ijk}$ depend on the service start time $(s_{jk})$ at customer $j$ by a vehicle $k$ (Figure 1); these costs are calculated as per Eq. (1), where $c_l$ is the unit penalty costs for any late arrival. It can be noted from the Figure 1 that the lateness is also bounded by $b'_i$.

\[ c'_{ijk} = \begin{cases} 
  c_{ij}, & \text{if } a_i \leq s_{jk} \leq b_j \\
  c_{ij} + c_l(s_{jk} - b_j), & \text{if } b_j < s_{jk} \leq b'_j 
\end{cases} \tag{1} \]

Now the VRPSTW can be mathematically formulated as

\[
\min \sum_{k \in K} \sum_{(i, j) \in A} c'_{ijk} x_{ijk} \tag{2}
\]

subject to

\[
\sum_{k \in K} \sum_{j \in V} x_{ijk} = 1 \quad \forall i \in C \tag{3}
\]

\[
\sum_{i \in C} d_i \sum_{j \in V} x_{ijk} \leq q \quad \forall k \in K \tag{4}
\]

\[
\sum_{j \in V} x_{0jk} = 1 \quad \forall k \in K \tag{5}
\]

\[
\sum_{i \in V} x_{ijk} - \sum_{j \in V} x_{hjk} = 0 \quad \forall h \in C, \quad \forall k \in K \tag{6}
\]

\[
\sum_{i \in V} x_{ijk} = 1 \quad \forall k \in K \tag{7}
\]

\[
s_k + t_{ij} - s_{jk} \leq (1 - x_{ijk})M \quad \forall (i, j) \in A, \quad \forall k \in K \tag{8}
\]

\[
a_i \leq s_k \leq b'_i \quad \forall i \in V, \quad \forall k \in K \tag{9}
\]

\[
x_{ijk} \in \{0, 1\} \quad \forall (i, j) \in A, \quad \forall k \in K \tag{10}
\]
The model contains two decision variables: $s_{jk}$ determines the service start time at customer $j$ as well as the travel cost of arc $(i, j)$, and $x_{ijk}$ represents whether arc $(i, j)$ is used in the solution ($x_{ijk} = 1$) or not ($x_{ijk} = 0$). The objective function (2) minimizes the total cost including the fixed cost for vehicle utilization and the travel cost on the arcs with late arrival penalty costs. Constraint (3) ensures that every customer is only serviced once, while constraint (4) is the capacity constraint that keeps the cumulative demands along a vehicle’s route within its capacity. Constraints (5), (6) and (7) are flow conservation constraints, specifying that every route shall start from and end at depot; while, on the route, if a vehicle travels to a customer location $h$ it must also travel from it. Constraint (8) is a time window constraint specifying that if a vehicle travels from $i$ to $j$, service at $j$ cannot start earlier than that at $i$. Here, $M$ is a large constant. Constraint (9) represents the relaxed time windows for the VRPSSTW and restricts the service start time at all vertices within their relaxed time windows $[a_i, b_i]$. Finally, constraint (10) ensures the integrality of the flow variables $x_{ijk}$.

### 3.2 Dantzig-Wolfe Decomposition of the VRPSTW

As it is clear from the formulation of the VRPSTW given in previous section, only constraint (3) couples all vehicles together while the rest of the constraints are separable for each vehicle. The Dantzig-Wolfe decomposition or commonly known column generation scheme decomposes the VRPSTW formulation in a set partitioning problem (Eq. (11)-Eq. (13)) by taking the coupling constraint (i.e. Eq. (3)) in it. Remaining constraints (Eq. (4)- Eq. (10)) formulate an Elementary Shortest Path Problem with Resource Constraint with Late Arrival Penalties (ESPPRCLAP) as a subproblem, with the objective function that minimizes the reduced cost of a path (Eq. (14)) starting and ending at the depot (i.e. a single vehicle route), called a column in the column generation scheme. The ESPPRCLAP is solved on the same graph as the VRPSTW, taking reduced cost on its arcs. The reduced cost of an arc $(i, j)$ (Eq. (15)) is obtained using the corresponding dual variable value $\pi_i$ (price or shadow cost) for each customer $i \in C$, obtained in the solution of the set partitioning master problem.

$$\min \sum_{p \in P} c_p y_p$$  \hspace{1cm} (11)

subject to

$$\sum_{p \in P} a_{ip} y_p = 1 \quad \forall i \in C$$ \hspace{1cm} (12)

$$y_p \in \{0, 1\} \quad \forall p \in P$$ \hspace{1cm} (13)

$$\min \sum_{(i,j) \in d} c'_{ij} x_{ij}$$ \hspace{1cm} (14)

subject to constraints (4) - (10) for each vehicle

$$c'_{ij} = c'_{ij} - \pi_i \quad \forall i \in C$$ \hspace{1cm} (15)

The role of the subproblem is to provide feasible single vehicle routes (also called paths or columns) to the linear program (LP) of the set partitioning master problem. Set $P$ is the set of all such feasible routes. The variable $y_p$ takes value 1 if the route $p \in P$, is selected and 0 otherwise. The cost of route $p$ is denoted by $c_p$, and $a_{ip}$ represents the number of times route $p$ serves customer $i$. The objective (Eq. (11)) selects a minimum cost set of routes from $P$ such that each customer $i \in C$ is serviced exactly once as per the constraint represented by Eq. (12). The integrality constraint is represented by Eq. (13); usually a linear programming
relaxation of this constraint is solved where $y_p$ can vary anywhere between 0 and 1. for more details, please refer to our earlier work in Qureshi et al., 2009.

4. THE DYNAMIC VRPSTW

4.1 General Framework
The proposed Dynamic Vehicle Routing and scheduling Problem with Soft Time Windows (D-VRPSTW), only considers the travel time uncertainty, therefore, all the remaining information such as customers' locations and demands are assumed to be known and fixed. The D-VRPSTW is modeled using the rolling horizon scheme, in which the complete scheduling horizon is divided into various time slots, each representing a time-based scenario. Thus initially, it can be defined on a complete Graph $G_{T_1}$ for the first time slot ($T_1$), which consists of all customers and with all vehicles ($k_i$) stationed at the central depot similar to the static VRPSTW (as shown in Figure 2(a)).

Figure 2. Representation of the dynamism of D-VRPSTW in rolling horizon scheme
The routes for the time slot $T_1$, would be planned as per the average travel times. No divergence is allowed once a vehicle leaves to visit a customer, i.e. the first customer on the route of an en-route vehicle is fixed. The time slots are marked with vehicle-based events, which means a new time slot is initiated as soon as any of the vehicle reaches the first customer on its route. It is possible to do the re-optimization at any or all of such events but it would be wasteful if no significant unexpected change has been observed in travel time in the previous time slot. This research would also consider the re-optimization only when some unexpected change is observed in the travel time on any arc, effectively reducing the number of re-optimization events, which however, have to coincide with a vehicle event.

With no diversion allowed, the locations of all vehicles and their times of availability are forecasted depending on their current activity. For example, a vehicle traveling towards a customer would be available after serving the demand of that particular customer. The Graph is then updated (as $G_{P_t}$ (Figure 2(b))) showing vehicle locations along with all customers except those which are either serviced or are the first customers of an en-route vehicle. The arcs in $G_{P_t}$ contain the updated travel times, therefore, routes planned in $T_2$ would be based on these updated travel time values. It may be noted that, nodes other than depot, which contain a vehicle resource, only have outbound links. A vehicle can abandon its remaining planned route and head back to the depot as well. This procedures goes on till the last customer is serviced (as shown by $G_{P_t}$ in Figure 2(c)). As the customers assigned to a particular vehicle may get changed at any of the route revision epoch (at the change of time slot), the proposed D-VRPSTW is suitable to either model a delivery system of single commodity such as heating oil, gasoline etc., or it can model any pick-up service.

4.2 Methodology
As the $G_{P_t}$ (Figure 1(b)) shows, the proposed framework leads to a Multi depot-Dynamic Vehicle Routing and scheduling Problem with Soft Time Windows (MD-VRPSTW) as each node with a vehicle resource would be considered as a virtual depot. Therefore, the proposed column generation based exact approach must incorporate the multi depot aspect as well. The column generation based approach developed for the static VRPSTW by authors (Qureshi et al., 2009) is extended to solve the D-VRPSTW by formulating a separate ESPPRCLAP subproblem for each of the virtual depot. At each column generation iteration, the set partitioning master problem receives columns (routes/paths of negative reduced costs) from all subproblems and then it optimizes the complete problem by selecting the best set of routes covering the demands of all the customers at hand. The dual variables' values (prices) obtained as a by-product, are then used to define new reduced cost matrices for each of the subproblems to generated new promising paths/columns, and the whole process is repeated till the subproblems fail to provide a negative reduced cost column. At this stage if the solution of the master problem LP is not integer, a branch and bound tree is explored. Therefore, at every route revision epoch, the column generation algorithm is embedded in a branch and price algorithm. In order to track and keep the total number of vehicles in the system, a new branching scheme has been introduced which first ensures an integer number of vehicles departing from each depot and then aims at integer flow variables. Figure 3, shows the flow chart of the complete column generation based algorithm for the D-VRPSTW.
5. TEST INSTANCE AND SCENARIOS

The performance of the D-VRPSTW concept and its column generation-based exact solution was evaluated using the Solomon’s benchmark instance (Solomon, 1987) called the R101-100 instance. The complete instance consists of 100 customers located randomly in a 100 x 100 space with random time windows; however, smaller instances can be constructed by selecting the first few customers. This study uses the first 25 customers instance (shown in Figure 4) which is called the R101-25-STW instance emphasizing the use of soft time windows. The scheduling horizon of the R101-25-STW instance is [0, 230] (i.e. the time windows at the depot node). The Euclidean distance is considered between vertices and by assuming a unit speed the travel time is also represented by the same value. A service time of 10 units is required at each customer to service its demand. Here we consider all time units as minutes.
The exact VRPSTW (static) solution of the R101-25-STW with $b'_i - b_i = 10$ minutes for all $i \in C$, results in seven vehicle routes as shown in Figure 5; whereas, Table 1 gives some details of this solution such as the ready time (time of availability) of each vehicle at the first customer on its route and the residual capacity of the vehicle (other details such as cost etc. are given in §6). At every event in the dynamic case, it is assumed that additional vehicles are available at the origin depot (vertex 1, shown as square in Figure 4) as well; during the re-optimization a new route can start from the origin depot if the solution with additional vehicle results in the least cost.

In order to evaluate the performance of the D-VRPSTW and to show its advantage two different cases were tested, both occurring at an event time of 32 minutes, i.e. at the occurrence of first vehicle arrival event at customer 15. Only two vehicles (No. 2 and 4) would have departed from the origin depot and would be available at customer 15 and 6 at 32 minutes and 34 minutes, respectively, while the rest of the vehicles will be available at the origin depot at 32 minutes.

Case1: The travel time along the arc (15, 16) is unexpectedly increased by 20 minutes due to any incident. It should be noted that customer 16 is the next customer on the route of vehicle no. 2 after the customer 15. At the event time of 32 minutes the vehicle would be available at the customer 15.
Case2: The travel time along the arc (3, 22) is unexpectedly increased by 20 minutes due to any incident. It should be noted that vertex 22 is the next vertex on the route of vehicle no. 3 after the vertex 3. At the event time of 32 minutes the vehicle would not have departed the origin depot and hence it would be available at the origin depot with all customers on its route available for re-optimization. It is important to note that in case of static version it is assumed that the change in travel time value is not known until the vehicle starts it travel on the affected arc (in this case arc (3, 22)), therefore in static VRPSTW, the change would only be known at time 50 minutes when the vehicle no. 2 is ready to leave customer no. 3 (Table 1).

![Figure 5 Routes of the initial static VRPSTW solution of the R101-25-STW instance](image-url)

<table>
<thead>
<tr>
<th>Vehicle</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Customer</td>
<td>13</td>
<td>15</td>
<td>3</td>
<td>6</td>
<td>24</td>
<td>12</td>
<td>8</td>
</tr>
<tr>
<td>Residual Capacity</td>
<td>181</td>
<td>180</td>
<td>193</td>
<td>174</td>
<td>171</td>
<td>188</td>
<td>195</td>
</tr>
<tr>
<td>Ready Time</td>
<td>63</td>
<td>32</td>
<td>50</td>
<td>34</td>
<td>68</td>
<td>67</td>
<td>81</td>
</tr>
</tbody>
</table>
6. COMPUTATIONAL RESULTS AND DISCUSSION

The algorithms were implemented in MATLAB, and were run on a computer with 2.67 GHz Intel Core i7 CPU with 6 GB of RAM. The vehicle operation cost (VOC) of 14.02 yen/minute is taken; while the fixed cost for a vehicle is set to 10417.5 yen. The unit late arrival penalty is taken as five times that of the VOC. These unit cost values are based on a survey of Japanese logistics companies and most commonly used in the city logistics-related literature (for example, see Taniguchi et al., 2001; Yamada et al., 2004; Ando and Taniguchi, 2006; Duin et al., 2007).

Figures 6 and 7 give the re-optimized routes obtained in the D-VRPSTW exact solutions for case 1 and case 2, respectively. It can be noted that in both cases, not only the affected arcs are no longer traveled (i.e. the arc (15, 16) in case 1 and arc (3, 22) in case 2), some changes in the other routes have also resulted in order to optimize the overall solution.

Figure 6 Re-optimized routes of the D-VRPSTW solution of the R101-25-STW instance in case 1
Figure 7 Re-optimized routes of the D-VRPSTW solution of the R101-25-STW instance in case 2

Table 2 gives the details of exact solution of the initial static VRPSTW (shown in Figure 5), its application in case 1 and case 2 (i.e. VRPSTW (static no change), rows no. 3 and 6, respectively) with no or minimal route changes. It also shows the re-optimized routes in case 1 and case 2 in the exact solution of the D-VRPSTW (rows no. 4 and 7, respectively). Column (3) shows the number of vehicle routes in each solution; whereas, Col.(4) and Col.(6) gives the total cost and the late arrival time (LAT) in these solutions. Differences in cost and late arrival time between the initial static solution and the static and dynamic solutions under case 1 and case 2, are reported in Col. (5) and Col. (7), respectively. All solutions contain seven vehicle routes, except the static solution in case 2 (with eight vehicles) due to the fact that the increase in travel time of arc (3, 22) makes the customer no. 22 infeasible to reach from any other customer, except customers 15 and 6, which are already served at the time vehicle no. 3 prepares to depart for customer 22 at time 50 minutes (Table 1). Therefore, at this instant the static solution would become infeasible and a change has to be made in the routing plan. Here we have considered a simple strategy that calls back the vehicle no. 3 from the customer 3 to depot and sends a new vehicle that completes the route allocated to vehicle no. 3 starting from the customer 22. It should be noted that the re-optimized solution of the D-VRPSTW also resulted in a route starting from customer 22 but on the same time the vehicle no. 3 is re-routed while shuffling the customers on other vehicles as well (Figure 7). As shown in Col.
(5), if we continue to use the static VRPSTW solution under the dynamic travel time scenarios (case 1 and case 2) high additional costs are incurred as compared to the D-VRPSTW. In case 1, use of D-VRPSTW saves about 855 Yen; whereas, the savings in case 2 are more significant (10577 Yen) due to the fact that the D-VRPSTW exact approach efficiently re-optimizes the whole solution and no additional vehicle is used. It is also significant to note that both D-VRPSTW exact solutions were obtained in less than a second in case 1 and case 2, which shows the efficiency of the column generation-based exact solution approach, specially for these small problems; however, it is expected that the computation time would grow rapidly for larger instance (say for 50 customer instances) as noted in the exact solution approach for the static VRPSTW (Qureshi et al., 2009).

<table>
<thead>
<tr>
<th>Case</th>
<th>Solutions</th>
<th>Veh.</th>
<th>Cost (Yen)</th>
<th>Difference from initial Solution (Yen)</th>
<th>LAT (minutes)</th>
<th>Difference from initial Solution (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td>VRPSTW (Static)</td>
<td>7</td>
<td>81901</td>
<td>-</td>
<td>7</td>
<td>-</td>
</tr>
<tr>
<td>Case 1</td>
<td>VRPSTW (Static, no change)</td>
<td>7</td>
<td>82952</td>
<td>1052</td>
<td>18</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>D-VRPSTW</td>
<td>7</td>
<td>82097</td>
<td>196</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td><strong>Benefit of D-VRPSTW</strong></td>
<td></td>
<td></td>
<td><strong>855</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case 2</td>
<td>VRPSTW (Static, no change)</td>
<td>8</td>
<td>92677</td>
<td>10776</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>D-VRPSTW</td>
<td>7</td>
<td>82100</td>
<td>199</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td><strong>Benefit of D-VRPSTW</strong></td>
<td></td>
<td></td>
<td><strong>10577</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Veh. = No. of vehicle routes, LAT = late arrival time

7. CONCLUSIONS AND FUTURE RESEARCH

Unexpected changes in traffic conditions and incidents results in the dynamic travel times on the roads of an urban areas. If the operations of a logistics company are planned, by ignoring this reality, it may affect the distribution or pick-up routes resulting in unexpected high cost and long delays. To cope with this situation, an exact solution approach for the Dynamic Vehicle Routing Problem with Soft Time Windows (D-VRPSTW) was presented in this paper. Its performance was evaluated under different dynamic travel time scenarios and it was observed that the D-VRPSTW results in significant cost savings.

Both case 1 and case 2 were obtained by arbitrarily adding an unexpected additional travel time due to any incident affecting only one arc. In reality, if one arc is affected it might impact the travel times on other arcs as well. Therefore in future research, it is planned to use some real life ITS-based data sets such as VICS data in the test instances.
REFERENCE


2003-55, GERAD.