ELEMENTARY SHORTEST PATH PROBLEM WITH RESOURCE CONSTRAINTS AND TIME DEPENDENT LATE ARRIVAL PENALTIES

Ali Gul QURESHI¹, Eiichi TANIGUCHI² and Tadashi YAMADA³

¹Student Member of JSCE, Doctoral Student, Faculty of Engineering, Kyoto University
(C-1 Kyotodaigaku Katsura, Nishikyo, Kyoto 615-8540)
E-mail: aligul@kiban.kuciv.kyoto-u.ac.jp
²Fellow Member of JSCE, Professor, Faculty of Engineering, Kyoto University
(C-1 Kyotodaigaku Katsura, Nishikyo, Kyoto 615-8540)
E-mail: taniguchi@kiban.kuciv.kyoto-u.ac.jp
³Full Member of JSCE, Associate Professor, Faculty of Engineering, Kyoto University
(C-1 Kyotodaigaku Katsura, Nishikyo, Kyoto 615-8540)
E-mail: t.yamada@kiban.kuciv.kyoto-u.ac.jp

Shortest Path Problem (SPP) has many Civil Engineering particularly Transportation Engineering applications. This study considers a variant of SPP; the Elementary Shortest Path Problem with Resource Constraints and Late Arrival Penalties (ESPPRCLAP). Time and capacity are considered as the resources and any delay violation of time window causes a late arrival penalty. An application of ESPPRCLAP is given in relation with Vehicle Routing and scheduling Problem with Time Windows (VRPTW), which is used as a principal tool to evaluate many city logistics measures such as route optimization, to mitigate typical problems caused by urban goods movement. Finally benefits of incorporating late arrival penalties has been shown by using ESPPRCLAP as subproblem in column generation solution of VRPTW based on data derived from practical road network.

Key Words: shortest path problem, city logistics, vehicle routing with soft time windows, column generation

1. INTRODUCTION

Shortest path problem (SPP) is used in many civil engineering related fields such as Construction Management and Planning, Environment Engineering and Transportation Engineering etc. City logistics¹ is a branch of civil engineering which deals with the urban freight related problems such as traffic congestion, loading and unloading on street side and environmental pollution, and their solutions. Measures such as route optimization, ideal location of logistics terminals and depots, load factor controls and cooperative delivery systems etc. are used for mitigating above-mentioned problems. An elementary version of SPP lies at the subproblem level of the exact solution of Vehicle Routing and scheduling Problem (VRP) which is used as a principal tool for evaluating many city logistics schemes. Route optimization not only serves the logistics firms, it also advances the objectives of the other stakeholders i.e. city administration, residents and customers. Route optimization would result in the least possible number of vehicle required to serve all the demands, traveling as minimum a distance as possible and decreasing the idling time of the vehicles. This would result in less pressure on the road network resources and less automobile related environmental problems. Thus, administrators get less traffic related problems, resident would get cleaner environment and customers would get faster deliveries. With the rapid increase in computer technology, the network handling capability and thus the importance of exact methods in vehicle routing is also increasing.

2. LITERATURE REVIEW

Many variants of SPP have found their way in research and applications such as Weight Con-
strained Shortest Path Problem (WCSPP)\(^2\); weight (demand) at every vertex is defined and a global weight (capacity) constraint is introduced and the shortest path can not accumulate more weights than the limiting weight \(W\) (capacity \(Q\)). Shortest Path Problem with Time Windows (SPPTW)\(^3,4\); either global or local time windows constraints are possible. In case of local time windows constraint, a path is feasible if it starts service at every vertex it visits, within its specified time windows \([a_i, b_i]\), where \(a_i\) and \(b_i\) show the earliest and latest service start times. Global time window constraint just requires that the path shall end within the given scheduling time or in other words specifying a time window of \([0, T]\) at the sink vertex, where \(T\) represents the maximum scheduling time. Shortest Path Problem with Resource Constraint (SPPRC)\(^5\) considers capacity as global constraint and time windows as local constraints. Depending on the cycling of the vertices in the shortest path once again many variants got attention of researchers, such as shortest path problem with 2-cycle elimination\(^6\), which does not allow cycling of the order \(i-j-i\) in the path. Similarly, shortest path problem with \(k\)-cycle elimination\(^7\) restricts cycling within a length of \(k\) elements in the path, while no repetition of the vertices is allowed in elementary shortest path problem\(^8,9\).

Most of the times Hard Time Windows (HTW) have been considered in SPPTW and in SPPRC\(^10,11\). As described above, in HTW service start at every vertex along the path must be within the specified time windows \([a_i, b_i]\), while waiting is allowed if path arrives before \(a_i\) (Fig. 1). On the other hand, Soft Time Windows (STW) are defined in two ways. In the first type, time windows \([a_i, b_i]\) defined by customers are not violated but an optimum service start time is searched by allowing waiting within the time windows. In the second type, both early and late arrival outside the customer defined time windows are allowed with suitable penalty costs. Many researchers have focused on the schedule optimization of a fixed path\(^12,13\) using STW-type1, the only work on the shortest path problem with these STW settings is due to Ioachim et al.\(^14\). The above mentioned references have considered the STW by minimizing a cost function at the vertices that depends on the service start time which is strictly within the time windows. The optimum service start time is found by allowing waiting within the time windows as well.

To the contrary, we considered the more general definition of STW-type2 by relaxing the latest possible arrival time \(b_i\) to \(b_i'\) i.e. a path can start service even if it arrives after the \(b_i\) (but not after \(b_i'\)) by paying a late arrival penalty cost. Furthermore, waiting is only allowed if the path arrives earlier than the service start time \(a_i\) i.e. no waiting is allowed within the relaxed time windows \([a_i, b_i']\). As waiting is allowed at no cost (Fig. 2), we defined these setting as Semi Soft Time Windows (STSW). As far as authors’ knowledge is concerned no SPPRC algorithm is available for above mentioned STW settings. However many researchers\(^15,16\) have considered those along with early arrival penalties in the heuristic solutions of Vehicle Routing and scheduling with Soft Time Windows (VRPSTW) including some practical applications\(^17\).

Such type of time windows is very important from the practical point of view as, while not considering waiting on early arrival as a penalty many logistics managers try to avoid the late arrival penalties, which affect the reliability of the logistics firms but on the same time could prove to be more economical.

Figs. 1 and 2 show the penalty cost functions of SPPRC and Shortest Path Problem with Resource Constraint and Late Arrival Penalties (SPPRCLAP), respectively. A very high penalty cost outside the time windows is used to model the hard time windows. Relaxed time windows are considered in SPPRCLAP (Fig. 2). The limit \(b_i'\) at every vertex \(i\) could be defined such that if the path travels the arc \((i, n+1)\) it still remains feasible i.e. it obeys \([a_{n+1}, b_{n+1}]\) where \(n+1\) represents the sink vertex.

![Fig.1 Penalty function for ESPPRC (HTW)](image1)

![Fig.2 Penalty function for ESPPRCLAP (STSW)](image2)
Note that time window at the sink vertex is not extended i.e. the path still follows the global time window constraint or the maximum scheduling time. The above feasibility condition produce very large relaxed time windows depending on how different \(b_i\) and \(b_i'\) are. The complexity of the labeling algorithm depends on the width of time windows (Desrochers et al.\(^5\)), therefore, we relaxed the latest possible service start times by 10 units only (equivalent to service time at every vertex). Table 1 summarizes the literature review about various versions of shortest path problem. Appendix A provides the list of abbreviations used in this paper.

This paper presents the model formulation and algorithm for Elementary Shortest Path Problem with Resource Constraint and Late Arrival Penalties (ESPPRC) by suitably modifying and extending the classical Elementary Shortest Path Problem with Resource Constraint (ESPPRCLAP) to consider time dependent late arrival penalties. Finally, we compared the relative benefits of ESPPRC by giving an application in column generation solution of Vehicle Routing and scheduling Problem with Time Windows (VRPTW) on an instance derived from real life logistics problem.

### Table 1 Summary of literature review about shortest path problem

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full name</th>
<th>Time windows</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>WCSPP</td>
<td>Weight Constrained Shortest Path Problem</td>
<td>-</td>
<td>Rail road management and long-haul aircraft maintenance (cited in Dumitrescu and Boland(^2))</td>
</tr>
<tr>
<td>SPPTW</td>
<td>Shortest Path Problem with Time Windows</td>
<td>HTW</td>
<td>Subproblem in school bus routing (VRPHTW(^3)) and in VRPHTW(^4) Schedule optimization of fixed path(^12,13) and Vehicle routing, aircraft assignment and crew scheduling(^14)</td>
</tr>
<tr>
<td>SPPRC</td>
<td>Shortest Path Problem with Resource Constraint</td>
<td>HTW</td>
<td>Subproblem in VRPHTW(^2)</td>
</tr>
<tr>
<td>SPPRC-2 cycle elimination</td>
<td>Shortest Path Problem with Resource Constraint with 2 cycle elimination</td>
<td>HTW</td>
<td>Subproblem in VRPHTW(^9,10)</td>
</tr>
<tr>
<td>SPPRC-k cycle elimination</td>
<td>Shortest Path Problem with Resource Constraint with k cycle elimination</td>
<td>HTW</td>
<td>Subproblem in VRPHTW (^1)</td>
</tr>
<tr>
<td>ESPPRC</td>
<td>Elementary Shortest Path Problem with Resource Constraint</td>
<td>HTW</td>
<td>Subproblem in VRPHTW(^8,9)</td>
</tr>
<tr>
<td>ESPPRC-2LAP</td>
<td>Elementary Shortest Path Problem with Resource Constraint and Late Arrival Penalties</td>
<td>SSTW</td>
<td>Subproblem in VRPSSTW (new concept of this paper)</td>
</tr>
</tbody>
</table>

3. MODEL FORMULATION

Consider a directed graph \(G = (V, A)\). The vertex set \(V\) includes the origin depot vertex 0 and destination depot \(n+1\), and the set of intermediate vertices \(N = \{1, 2, \ldots, n\}\). These intermediate vertices represent customer locations in logistics terms.

The arc set \(A\) consists of all the feasible arcs \((i, j), i, j \in V\). Both \(c_{ij}\) and \(t_{ij}\) are associated with each arc \((i, j) \in A\), no arc terminates in vertex 0 and no arc originates from vertex \(n+1\). Time \(t_{ij}\) includes the travel time on arc \((i, j)\) and service time at vertex \(i\). With every vertex of \(V\) associated a demand \(d_i\), with \(d_0 = d_{n+1} = 0\), and a time window \([a_i, b_i]\) representing the earliest and the latest possible service start times.

This study incorporates the semi-soft time windows constraint by extending the customer defined latest possible service (delivery or pickup of customer goods) start time \(b_i\) to relaxed latest possible service start time \(b_i'\) as shown in Fig. 2. If \(s_j\) defines the service start time by a path or vehicle at a customer vertex \(j \in N\), the modified time dependent travel cost, \(c_{ij}'\) was defined as a function (eq. (1)) of \(s_j\) for all arcs \((i, j) \in A\) where \(c_i\) represents the unit late arrival penalty. Furthermore, waiting is only allowed if the path arrives at the vertex before its service start time. As discussed earlier, soft time windows research on SPPTW mostly allowed
waiting in the time windows and did not allow start of service outside the time windows.  

\[ c'_{ij} = \begin{cases} 
    c_{ij}, & \text{if } s_j \leq b_j \\
    c_{ij} + c_i(s_j - b_j), & \text{if } s_j > b_j 
\end{cases} \quad (1) \]

ESPPRCLAP aims at finding the least cost path for a single vehicle starting and finishing at depot, intermediate visiting as many customers as possible, as long as the resource constraints are not violated. Being elementary any customer can only be visited once along the path. The two resources taken are weight (capacity of vehicle) and time. Capacity resource constraint is a global constraint that limits the accumulated customer demands (in terms of weights to be delivered or picked up) to a pre-specified accumulated weight limit \( q \) (i.e., vehicle capacity). Time resource constraint acts both at global and local level; globally path should start and end within time windows of depot \([a_0, b_0]\) and at the local level service at every visited customer should be within relaxed time windows at each customer vertex (i.e. \([a_i, b'_i]\)). A path departs the origin depot to suit the arrival constraint \((a_i)\) of the first customer it visits. Using the arrival time dependent travel cost and above mentioned conditions ESPPRCLAP can be formulated as follows:

\[
\text{min} \sum_{(i,j) \in A} c'_{ij}x_{ij} \quad (2)
\]

subject to

\[
\sum_{i \in V} d_{ij} \sum_{j' \in V} x_{ij} \leq q \quad (3)
\]

\[
\sum_{j \in V} x_{ij} = 1 \quad (4)
\]

\[
\sum_{j' \in V} x_{ij} - \sum_{j \in V} x_{ij} = 0 \quad \forall \; h \in N \quad (5)
\]

\[
\sum_{j \in V} x_{ij} = 1 \quad (6)
\]

\[
s_i + t_{ij} - s_j \leq (1 - x_{ij})M_{ij} \quad \forall \; (i,j) \in A \quad (7)
\]

\[
a_i \leq s_i \leq b'_i \quad \forall \; i \in V \quad (8)
\]

\[
x_{ij} \in \{0,1\} \quad \forall \; (i,j) \in A \quad (9)
\]

The model contains two decision variables \( s_j \) that determines the travel cost of arc \((i,j)\), and \( x_{ij} \) which determines whether arc \((i,j)\) is used in the solution \((x_{ij} = 1)\) or not \((x_{ij} = 0)\). Objective equation (2) minimizes the total cost of the solution including the travel cost on the arcs as well as any late arrival penalty cost. \( M_{ij} \) is a big constant\(^{18}\) for each arc \((i,j)\). Constraint (3) is capacity constraint where \( q \) denotes the maximum limit of the weight (i.e., customer demands) that can be accumulated along the path. Constraints (4), (5) and (6) are flow conservation constraints. Constraint (7) is time windows constraint specifying that if arc \((i,j)\) is used, service at \(j\) cannot start earlier than that at \(i\). Constraint (8) specifies the relaxed time windows for ESPPRCLAP and restricts the service start time at all vertices must be within their relaxed time windows \([a_i, b'_i]\). Note that the constraint (7) is the linearized form of the original non-linear constraint (10).

\[
x_{ij}(s_j + t_{ij} - s_i) \leq 0 \quad \forall \; (i,j) \in A \quad (10)
\]

Note that if we just change the constraint (8) to (11) we get the classical ESPPRC i.e. with hard time windows not allowing any violations of time windows.

\[
a_i \leq s_i \leq b_i \quad \forall \; i \in V \quad (11)
\]

The feasible arcs set follow the inequality \( a_i + t_{ij} \leq b'_i \) instead of \( a_i + t_{ij} \leq b_i \). This resulted in an extended network for ESPPRCLAP compared to ESPPRC network.

4. ESPPRCLAP ALGORITHM

The ESPPRC is a NP-hard problem in strong sense\(^{19}\), but there exists some pseudo-polynomial dynamic programming labeling algorithm for solving it. As ESPPRCLAP is more general and can be reduced or converted to ESPPRC by setting very high unit late arrival penalty cost \( c_i \) (such as \( c_i = \infty \)), it is also a NP-hard problem. The ESPPRCLAP algorithm is based on the template-labeling algorithm for shortest path problem described in Irnich and Villeneuve\(^7\) that incorporates the ideas of permanent labels presented in Desrochers and Soumis\(^3\). The labeling algorithm for the ESPPRCLAP is obtained by incorporating time dependent late arrival penalties and using \(|V|\) (where \(|V|\) shows the cardinality of set \( V \)) extra resources and \( S \) as number of unreachable vertices defined in Feillet et al.\(^8\). For the basic knowledge about labeling algorithms the readers are referred to Ahuja et al.\(^20\), while excellent reviews about SPPRC can be found in Irnich and Desaulniers\(^5\) and Irnich and Villeneuve\(^3\). The algorithm can be described in the following steps:

1. Define \( L = \{ \text{res}(L), \tau(L), q(L), c(L), \text{vis}(L), \text{pred}(L), S(L), \text{Label no.} \} \)
Where

\[
\begin{align*}
res(L) &= \text{resident vertex of the label } L \\
t(L) &= \text{time resource of the label, which shows the service start time at } res(L) \\
g(L) &= \text{demand resource of the label, which shows cumulative demand along the path ending at } res(L) \\
c(L) &= \text{cost of the label, which shows cost of the path ending at } res(L) \\
\text{vis}(L) &= \text{visited vertex resource vector of the label with } |V| \text{ entries of 1 for all unreachable vertices and 0 for all reachable vertices}, \\
pred(L) &= \text{immediate predecessor vertex of } \text{res}(L) \text{ on path } p(L) \text{ represented by the label} \\
S(L) &= \text{total number of unreachable vertices from } res(L) \\
\text{Label no.} &= \text{a label counter assigning a unique number to each label.}
\end{align*}
\]

2. Initialization

\[L = \{L\} \text{ be the set of all unprocessed labels starting with } L = \{L_0\}, \text{ with } res(L_0) = 0, \text{ i.e. the origin depot.} \]

\[U(i) = \Phi = \text{set of all useful labels for all } i \in V, \text{ where } \Phi = \text{empty set.} \]

3. Repeat steps 4, 5 and 6 until \(L \neq \Phi\)

4. Label Selection

Take the label \(L'\) with \(t(L')\) that is min. from the set of unprocessed labels \(L\) and let \(i = \text{res}(L')\), Set

\[L(i) = \{L \in L: \text{res}(L) = i \text{ and } t(L) < t(L') + \min(t L_0)\}. \]

Set \(L = L \setminus L(i)\).

5. Dominance Rule

Apply dominance rule to all labels in \(L(i)\) using \(U(i) \cup L(i)\). Let

\[U'(i) = \{L(i) | \text{ which survive the dominance rule }\}. \]

Set \(U(i) = U(i) \cup U'(i)\).

6. Path Extension

Outer loop: For each label \(L \in U'(i)\).

Inner Loop: For all vertices \(j\) having entry 0 in \(\text{vis}(L)\), extend \(L\) to all \(j\). Find \(s_j\) by \(\max \{a_j, t(L) + t_0\}\).

Calculate the cost by \(c(L) + c'_{ij}\) as per eq. (1).

Update demand and visited vertex resource vector.

If \(j = n+1\), i.e. depot, Add \(L\) to \(U(n+1)\). Else, add \(L\) to \(L\).

7. Path Filtration

The shortest path would be the one represented by the label with least cost among all the labels \(L \in U(n+1)\); to obtain the path it represents, link back the labels following the \(\text{pred}(L)\).

(1) Dominance rule

A labeling algorithm generates new states or labels from the previously generated labels, dominance rules are implemented to avoid proliferation of labels. As described in Feillet et al.8), in ESPPRC a label \(L_1\) dominates another label \(L_2\) (both residing at same vertex), if it satisfies the following dominance criterions:

\[
\begin{align*}
t(L_1) &\leq t(L_2) \quad \text{(12)} \\
g(L_1) &\leq g(L_2) \quad \text{(13)} \\
\text{vis}(L_1) &\leq \text{vis}(L_2) \quad \text{(14)} \\
c(L_1) &\leq c(L_2) \quad \text{(15)}
\end{align*}
\]

If all the abovementioned conditions hold, then \(L_2\) is deleted as any future extensions of \(L_2\) are also possible by extensions of \(L_1\) (eq. (14)) at less resource consumption (eq. (12) and (13)) and at less cost (eq. (15)). Note that \(S(L)\) is not used in the dominance rule actually it helps to accelerate it as condition (14) is only possible if \(S(L_1) \leq S(L_2)\). As waiting up to \(a_i\) is allowed at no cost if the path arrives before \(a_i\) and the cost matrix still follows the triangular inequality, i.e. \(c_{ij} \leq c_{ik} + c_{kj}\) because the penalty function is non-decreasing, same dominance rules are also applicable for ESPPRC-CLAP.

(2) Path extension

During the path extension step an existing state at vertex \(i\) is extended to all new possible states by updating the resource consumptions and cost. If a state \(L_i\) with \(\text{res}(L_i) = i\) is to be extended to \(L_j\) with \(\text{res}(L_j) = j\) using arc \((i, j)\) following rules are used to update the resources:

\[
\begin{align*}
t(L_j) &= \max \{t(L_i) + t_{ij}, a_j\} \quad \text{(16)} \\
g(L_j) &= g(L_i) + d_j \quad \text{(17)} \\
c(L_j) &= \begin{cases} 
 c(L_i) + c_{ij}, & \text{if } t(L_j) \leq b_j \quad \text{(18)} \\
 c(L_i) + c_{ij} + c_i(t(L_i) - b_j), & \text{if } t(L_j) > b_j
\end{cases}
\end{align*}
\]

A newly generated label inherits the visited vertex resource vector from its processor label. This
vector is further updated by assigning a value of 1 for the vertices $k$ having a 0 entry in $vis(L_i)$ and which become unreachable from new resident vertex $j$ due to violation of any of the following equations:

$$t(L_j) + t_{jk} > b'_j$$

$$q(L_j) + d_k > q$$

Finally, the total number of unreachable nodes $S(L_j)$ in the new state is calculated as the sum of $vis(L)$ and $i$ is set as the immediate predecessor in $pred(L)$. New path extension rules were defined for ESPPRCLAP to incorporate the variable costs (due to possible late arrival penalty) and in $vis(L_j)$ update rule to consider relaxed time windows $[a_i, b_i]$ as shown in eq. (18) and eq. (19).

5. ESPPRCLAP AS A SUBPROBLEM

Vehicle Routing and scheduling Problem with Time Windows (VRPTW) is a well known NP-hard problem which consists of determining a set of optimum routes covering all the demands of a given set of customers without violating the capacity of used vehicles and with start of service at each customer $i$ within its pre-specified time window $[a_i, b_i]$. Considering the nature of time windows to be hard or soft two variants are formed i.e. Vehicle Routing and scheduling Problem with Hard Time Windows (VRPHTW), and Vehicle Routing and scheduling Problem with Soft Time Windows (VRPSTW).

Elementary Shortest Path Problem with Resource Constraints (ESPPRC) appears as the subproblem in Dantzig-Wolfe decomposition of VRPTW with the set partitioning master problem. Many researchers have worked with various shortest path variations as subproblem for instance, Desrochers et al.\textsuperscript{10} used 2-cycle elimination while solving the relaxed shortest path subproblem. Irnich and Villeneuve\textsuperscript{7} used k-cycle elimination with $k>3$. Recently, ESPPRC has been solved as the subproblem\textsuperscript{9, 9}. Feillet et al.\textsuperscript{8} also showed that when used as subproblem, ESPPRC gives the tightest lower bound in most of the instances. An exhaustive review about VRPHTW can be found in Cordeau et al.\textsuperscript{18}. The ESPPRCLAP has been used in this study as the subproblem resulting in the Dantzig-Wolfe decomposition of Vehicle Routing and scheduling Problem with Time Windows with Semi Soft Time Windows (VRPSSSTW), considering the late arrival penalties only, and not penalizing early arrival (i.e. waiting is allowed at no cost).

(1) Column generation

Working as the subproblem ESPPRCLAP gives feasible shortest paths subjected to the constraints (3)-(9). The master problem consists of selecting a set of paths to minimize the cost and ensure that all the customers are serviced exactly once and by one vehicle. As vehicles are assigned paths generated in the subproblem ESPPRCLAP, they depart the depot so that arrival time at the first customer on their path matches with its earliest service start time $a_i$. Therefore all vehicles may not leave at the same time. Mathematically master problem is described as:

$$\min \sum_{p \in P} c_p y_p$$

subject to

$$\sum_{p \in P} a_{i,p} y_p = 1 \quad \forall i \in C$$

$$y_p \in \{0,1\} \quad \forall p \in P$$

Where $y_p$ takes value 1 if the path $p$ is selected and 0 otherwise. The cost of the path $p$ is denoted by $c_p$, and $a_{i,p}$ represents the number of times path $p$ serves customer $i$. $P$ is the set of all feasible paths.

In actual application the set covering master problem is solved by replacing constraint (22) by (24), as linear programming relaxation of set covering type master problem is more stable than set partitioning type. Set covering constraint restricts the range of dual variables (i.e., prices) to only positive values, whereas in set portioning both positive as well as negative values are possible.

$$\sum_{p \in P} a_{i,p} y_p \geq 1 \quad \forall i \in C$$

Master problem generates the prices (value of dual variables) $\pi_i$, which are used to find the reduced cost for the arcs in ESPPRCLAP graph as follows:

$$c_{ij} = c_{ij} - \pi_i \quad \forall i \in V$$

To incorporate semi soft time windows, cost $c_{ij}'$ is calculated by the eq. (1) using $c_{ij}$ instead of $c_{ij}$.

6. TEST PROBLEMS

The performance of ESPPRCLAP is analyzed on VRPTW instance derived from Tokyo Metropolitan
Road Network data. The customer locations are a chain of convenience stores with their demands being known. Time windows are generated randomly and a fixed service time of 10 minutes is considered at every customer vertex. The width of time windows was fixed to 10 minutes at every customer node, while their earliest possible service start times $a_i$ were generated in the range of $[a_0 + t_0, b_0 - t_0]$.

Table 2 shows the demand (weights) and hypothetical time windows data for each vertex. The vehicle capacity was used as two tons (2000 kg).

<table>
<thead>
<tr>
<th>Vertex</th>
<th>Demand kg</th>
<th>$a_i$ minute</th>
<th>$b_i$ minute</th>
<th>$b_i'$ Minute</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>510</td>
<td>1440</td>
<td>1440</td>
</tr>
<tr>
<td>2</td>
<td>117</td>
<td>615</td>
<td>625</td>
<td>635</td>
</tr>
<tr>
<td>3</td>
<td>100</td>
<td>574</td>
<td>584</td>
<td>594</td>
</tr>
<tr>
<td>4</td>
<td>83</td>
<td>550</td>
<td>560</td>
<td>570</td>
</tr>
<tr>
<td>5</td>
<td>133</td>
<td>520</td>
<td>530</td>
<td>540</td>
</tr>
<tr>
<td>6</td>
<td>83</td>
<td>538</td>
<td>548</td>
<td>558</td>
</tr>
<tr>
<td>7</td>
<td>133</td>
<td>529</td>
<td>539</td>
<td>549</td>
</tr>
<tr>
<td>8</td>
<td>133</td>
<td>530</td>
<td>540</td>
<td>550</td>
</tr>
<tr>
<td>9</td>
<td>300</td>
<td>556</td>
<td>566</td>
<td>576</td>
</tr>
<tr>
<td>10</td>
<td>133</td>
<td>595</td>
<td>605</td>
<td>615</td>
</tr>
<tr>
<td>11</td>
<td>50</td>
<td>540</td>
<td>550</td>
<td>560</td>
</tr>
<tr>
<td>12</td>
<td>133</td>
<td>580</td>
<td>590</td>
<td>600</td>
</tr>
<tr>
<td>13</td>
<td>150</td>
<td>585</td>
<td>595</td>
<td>605</td>
</tr>
<tr>
<td>14</td>
<td>167</td>
<td>525</td>
<td>535</td>
<td>545</td>
</tr>
<tr>
<td>15</td>
<td>83</td>
<td>592</td>
<td>602</td>
<td>612</td>
</tr>
<tr>
<td>16</td>
<td>150</td>
<td>540</td>
<td>550</td>
<td>560</td>
</tr>
<tr>
<td>17</td>
<td>133</td>
<td>625</td>
<td>635</td>
<td>645</td>
</tr>
<tr>
<td>18</td>
<td>133</td>
<td>610</td>
<td>620</td>
<td>630</td>
</tr>
<tr>
<td>19</td>
<td>83</td>
<td>625</td>
<td>635</td>
<td>645</td>
</tr>
<tr>
<td>20</td>
<td>117</td>
<td>583</td>
<td>593</td>
<td>603</td>
</tr>
<tr>
<td>21</td>
<td>117</td>
<td>550</td>
<td>560</td>
<td>570</td>
</tr>
<tr>
<td>22</td>
<td>150</td>
<td>547</td>
<td>557</td>
<td>567</td>
</tr>
<tr>
<td>23</td>
<td>100</td>
<td>590</td>
<td>600</td>
<td>610</td>
</tr>
<tr>
<td>24</td>
<td>83</td>
<td>570</td>
<td>580</td>
<td>590</td>
</tr>
<tr>
<td>25</td>
<td>83</td>
<td>605</td>
<td>615</td>
<td>625</td>
</tr>
<tr>
<td>26</td>
<td>133</td>
<td>601</td>
<td>611</td>
<td>621</td>
</tr>
<tr>
<td>27</td>
<td>200</td>
<td>565</td>
<td>575</td>
<td>585</td>
</tr>
<tr>
<td>28</td>
<td>117</td>
<td>615</td>
<td>625</td>
<td>635</td>
</tr>
<tr>
<td>29</td>
<td>200</td>
<td>565</td>
<td>575</td>
<td>585</td>
</tr>
<tr>
<td>30</td>
<td>133</td>
<td>560</td>
<td>570</td>
<td>580</td>
</tr>
<tr>
<td>31</td>
<td>183</td>
<td>635</td>
<td>645</td>
<td>655</td>
</tr>
<tr>
<td>32</td>
<td>133</td>
<td>575</td>
<td>585</td>
<td>595</td>
</tr>
<tr>
<td>33</td>
<td>100</td>
<td>605</td>
<td>615</td>
<td>625</td>
</tr>
<tr>
<td>34</td>
<td>117</td>
<td>645</td>
<td>655</td>
<td>665</td>
</tr>
<tr>
<td>35</td>
<td>117</td>
<td>510</td>
<td>520</td>
<td>530</td>
</tr>
<tr>
<td>36</td>
<td>100</td>
<td>626</td>
<td>636</td>
<td>646</td>
</tr>
<tr>
<td>37</td>
<td>133</td>
<td>515</td>
<td>525</td>
<td>535</td>
</tr>
<tr>
<td>38</td>
<td>117</td>
<td>635</td>
<td>645</td>
<td>655</td>
</tr>
<tr>
<td>39</td>
<td>150</td>
<td>619</td>
<td>629</td>
<td>639</td>
</tr>
</tbody>
</table>

Fig. 3 Test instance road network and customer location
Time window at every vertex was extended by 10 minutes to allow late arrivals. A scaled cost matrix was used in programs taking VOC = 1, all the costs are also scaled to same level. At the end the results are scaled up again. We used same column generation algorithm settings for hard time windows as for semi soft time windows variant except the late arrival penalties and taking fixed cost of vehicle utilization in path costs $c_p$.

Fig. 4 shows the routes obtained in VRPHTW using ESPPRC as its subproblem, five vehicles were required to service all the customers. Fig. 5 shows the routes resulted in ESPPRCLAP application as subproblem in the VRPSSTW solution. Route for truck 2 in ESPPRCLAP covers few customers on routes 1 and 2 in ESPPRC, whereas route for truck 4 in ESPPRCLAP covers few customers on route 4 and 5 in ESPPRC. Thus by relaxing the latest possible service start times ESPPRCLAP was able to find longer path serving more customers therefore the optimum solution of VRPSSTW contains one less vehicle.

![Fig. 4](image1.png)

(a) Route for truck no. 1
(b) Route for truck no. 2
(c) Route for truck no. 3
(d) Route for truck no. 4
(e) Route for truck no. 5

Fig. 4 Routes of the VRPHTW solution using ESPPRC as subproblem (TD1_39_htw)
Figure 5: Routes of the VRPSSTW solution using ESPPRCLAP as subproblem (TD1_39_sstw)

Table 3 summarizes the results for the solution of VRPSSTW and VRPHTW using ESPPRCLAP and ESPPRC as their subproblems, respectively. The test instances are named as TD1_39_sstw and TD1_39_htw based on the nature of time windows to be semi soft or hard respectively as well as on the fact that they contain 39 vertices (1 depot and 38 customers). Col. 2 in Table 3 gives the number of feasible arcs, as discussed earlier the network size (number of feasible arcs) for ESPPRCLAP are considerably higher than ESPPRC. Col. 3 specifies the number of branch and bound nodes. Col. 4 gives the LP lower bound obtained at the root node of the branch and bound tree when the set partitioning LP stops due to its stopping criteria i.e. either subproblem returns no negative reduced cost column or if the simplex multipliers (prices) are same as previous iteration. Col. 5 shows the lower bound at the end of the column generation algorithm. Col. 6 gives the optimum integer solution obtained at the end of column generation algorithm for VRPSSTW this includes the fixed vehicle cost, travel cost of the used arcs as well as the late arrival penalty. Cols. 7 and 8 show the number of iteration, number of columns added to master problem LP for column generation algorithm. Col. 9 gives the average number of labels generated in each run of ESPPRCLAP.
Table 3 Summery of Exact Solutions of VRPSSTW using ESPPRCLAP and VRPHTW using ESPPRC

<table>
<thead>
<tr>
<th>Instance</th>
<th>Branch and Bound Nodes</th>
<th>Lower bound LB₁</th>
<th>Lower bound LB₂</th>
<th>Integer solution IP-opt</th>
<th>No. of subproblem iterations</th>
<th>Col. added to LP</th>
<th>Labels per subproblem iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>TD1_39_sstw*</td>
<td>782</td>
<td>12</td>
<td>3173.253</td>
<td>3253.2</td>
<td>3255.7</td>
<td>113</td>
<td>2348</td>
</tr>
<tr>
<td>TD1_39_htw**</td>
<td>707</td>
<td>11</td>
<td>172.3</td>
<td>176.6</td>
<td>176.8</td>
<td>81</td>
<td>278</td>
</tr>
</tbody>
</table>

Table 4 Cost Components of Exact Solutions of VRPSSTW and VRPHTW

<table>
<thead>
<tr>
<th>Instance</th>
<th>Waiting Time (minute)</th>
<th>Running Cost (Yen)</th>
<th>Late Arrival Penalty (Yen)</th>
<th>Total Cost (Yen)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TD1_39_sstw*</td>
<td>19.8</td>
<td>2743.71</td>
<td>1233.76</td>
<td>45647.5</td>
</tr>
<tr>
<td>TD1_39_htw**</td>
<td>119.5</td>
<td>2478.7</td>
<td>0</td>
<td>54566.2</td>
</tr>
</tbody>
</table>

* Tokyo Data 1 with 39 vertices and semi soft time windows
** Tokyo Data 1 with 39 vertices and hard time windows

Table 4 shows the solution cost details, Col. 2 gives the number of vehicle required, and Col. 3 shows the waiting time in the solution of VRPTW. Col. 4 gives the running cost and Col. 5 shows the total late arrival penalty. Finally, Col. 6 gives the total solution cost composed of fixed vehicle cost, running cost and late arrival penalty costs.

A comparison of the tables shows that ESPPRCLAP is more difficult to solve than ESPPRC. The average number of labels generated per iteration in ESPPRCLAP is much more (about seven times) than those in ESPPRC (Table 3, Col. 9). Allowing late arrival increases the number of feasible arcs (Table 3, Col. 2) which results in increase in the size of underlying network for ESPPRCLAP as in this case a 10.61 % increase was observed. Due to increase in network size and in the time windows width (by bi - bi units), the number of states or labels also increase that results in higher computation time for ESPPRCLAP. A bigger branch and bound tree requiring more subproblem iterations also contributes in this regard. An interesting aspect is the massive decease in waiting time (as much as by 83.4% = (119.5 - 19.8) / 119.5) along the paths produced by ESPPRCLAP as the VRPTW subproblem, even not considering it as an objective. This decrease in waiting is not problem specific as Qureshi et al. reports similar reduction when they used the presented STW settings in their heuristic analysis of VRPTW on Solomon’s test instances.

The main contribution of adding late arrival penalties was thought to reduce the number of vehicles by improving vehicle utilization and to save overall cost of the solution. As expected using ESPPRCLAP instead of ESPPRC as the subproblem of VRPTW reduced the number of vehicles from five to four (Table 4, Col. 2) and also saved overall solution cost (Table 4, Col. 6). One of the main purposes of this study was to analyze that this reduction is obtained on how much extra computational effort. As much as 16.6 % of the cost can be saved by allowing late arrivals with 4.89 times more computational effort. Though this might not seems attractive at the first glance but with advancement of computer technology, the exact solution techniques such as used in this paper run very fast. For example, VRPHTW solution using ESPPRC was found for this problem only in 106.52 seconds using ants routing data. Therefore a 16.6% saving in cost is really significant while considering extra computation effort of less than 9 minutes. Furthermore, the resulting decrease shows the net savings to a logistics firm only, not including the environment and other social benefits to other stakeholders of city logistics. Less number of vehicles and reduced waiting time advances the objectives of the city administration by improving traffic related problems such as congestion and on-street parking. Reduction in waiting time not only helps logistics firms by saving vehicle operating and idling cost, it helps the cause of city administrative and general public (residents of the area of service) by reducing environmentally non-friendly emissions as most of
the times delivery vehicles wait near the next client location to be served in engine on state.

The only suffering stakeholders could be the clients, but as can be seen in Table 4, Col. 5, late arrival penalties can be used to offer good concessionary packages to clients as well for any inconvenience is caused by late arrival. Moreover, few and not all time windows could be relaxed to satisfy either clients with very strict time windows or the ones which are very important to the logistics firm.

8. CONCLUSIONS

In this paper, we have considered time dependent late arrival penalties in the formulation of ESPPRC for the first time, resulting in ESPPRCLAP. Cost calculation and visiting resource vector in the path extension step were suitably modified to obtain a labeling algorithm for ESPPRCLAP. As an application column generation technique for VRPHTW was also extended for the VRPSSTW by solving ESPPRCLAP instead of ESPPRC. With relaxed time window it was able to produce longer paths serving more customers with fewer vehicles. This resulted in reduced number of required vehicles in VRPSSTW solution reducing the overall solution cost. Waiting time was also reduced remarkably as a byproduct. Reduced waiting time and less number of vehicles, not only effects the overall cost; it could be helpful in reducing traffic related problems such as on street parking, congestions and other environment related issues. Although use of ESPPRCLAP as subproblem was found to be costly as compared to ESPPRC in column generation solution of VRPTW as far as the computation time is concerned, the overall cost significantly reduced as compared to extra computational effort required.

ESPPRCLAP is a relaxation of ESPPRC, and thus cost savings depend on the fact that how severely hard time windows constrain the problem. Narrow time windows (tightly constrained problems) will result in better relaxed solution, while very wide time windows hardly constrain the problem. Thus, relaxed solution, at most, would lead to same solution cost as original, but with more computation requirements. A trivial lower bound on number of vehicles is obtained using capacity constraint i.e. sum of all customer demands divided by capacity of single vehicle. But time windows are the main factors determining the additional (to the trivial lower bound by capacity constraint) number of vehicles in ESPPRC and ESPPRCLAP. Furthermore, the slope of the penalty curve $c_l$ between $b_i$ and $b_i'$ also effects the solution quality in relaxed version as a very large value of $c_l$ (such as $c_l = \infty$) changes ESPPRCLAP to ESPPRC.

9. FUTURE RESEARCH

At present only static travel time is considered in the model, while in practice travel times vary dynamically in nature. Furthermore, large change in waiting time was observed even in considering only late arrival penalties. Therefore, future directions in this regard could be devising a suitable elementary shortest path problem which is able to take into account both early arrival and late arrival penalties as well as to accommodate variable travel times.

APPENDIX A LIST OF ABBREVIATIONS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESPPRC</td>
<td>Elementary Shortest Path Problem with Resource Constraint</td>
</tr>
<tr>
<td>ESPPRCLAP</td>
<td>Elementary Shortest Path Problem with Resource Constraint and Late Arrival Penalties</td>
</tr>
<tr>
<td>HTW</td>
<td>Hard Time Windows</td>
</tr>
<tr>
<td>SPP</td>
<td>Shortest path problem</td>
</tr>
<tr>
<td>SPPRC</td>
<td>Shortest Path Problem with Resource Constraint</td>
</tr>
<tr>
<td>SPPTW</td>
<td>Shortest Path Problem with Time Windows</td>
</tr>
<tr>
<td>SPPRCLAP</td>
<td>Shortest Path Problem with Resource Constraint and Late Arrival Penalties</td>
</tr>
<tr>
<td>SSTW</td>
<td>Semi Soft Time Windows</td>
</tr>
<tr>
<td>STW</td>
<td>Soft Time Windows</td>
</tr>
<tr>
<td>VOC</td>
<td>Vehicle Operating Cost</td>
</tr>
<tr>
<td>VRP</td>
<td>Vehicle Routing and scheduling Problem</td>
</tr>
<tr>
<td>VRPTW</td>
<td>Vehicle Routing and scheduling Problem with Time Windows</td>
</tr>
<tr>
<td>VRPHTW</td>
<td>Vehicle Routing and scheduling Problem with Hard Time Windows</td>
</tr>
<tr>
<td>VRPSSTW</td>
<td>Vehicle Routing and scheduling Problem with Time Windows with Semi Soft Time Windows</td>
</tr>
<tr>
<td>WCSPP</td>
<td>Weight Constrained Shortest Path Problem</td>
</tr>
</tbody>
</table>

REFERENCES


(Received July 6, 2007)