【土木計画学研究・論文集 Vol.26 no.4 2009年9月】

HYBRID INSERTION HEURISTICS FOR VEHICLE ROUTING PROBLEM WITH SOFT TIME WINDOWS

Ali Gul Qureshi **, Eiichi Taniguchi*** and Tadashi Yamada****

1. Introduction

Industrial growth and expanding employment opportunities have led to the urban-oriented economic development in many countries. Demand of transportation, in terms of passengers as well as of freight is also increasing in and around these big urban conurbations. A high proportion of total goods movement occurs within cities[1)], and most of this movement is based on road-based transport. Traffic congestion, noise, vibrations, generation of NOx, SPM, CO_2 and other environmental problems, crashes and loading and unloading on the street side are typical problems caused by the road-based freight transport in urban areas.

Such freight movement-related problems have magnified the need for researches in the field of city logistics. The Vehicle Routing and scheduling Problem with Time Windows (VRPTW) can be used as a tool for evaluating many city logistics schemes. For example, the VRPTW can be used in the analysis of cooperative delivery systems[2)] and ideal location of logistics terminals[3)], which belong to infrastructure planning and management problems in city logistics. Depending on the nature of the time windows, the VRPTW is further expanded to the Vehicle Routing and scheduling Problem with Hard Time Windows (VRPHTW) and the Vehicle Routing and scheduling Problem with Soft Time Windows (VRPSTW).

Most of the exact optimization research has been directed towards the hard time windows variant, though it lacks the practicality found in real life problems. Recently column generation techniques have been used efficiently for the VRPHTW, successfully solving large size problems as well as significantly reducing the required computational efforts. For hard time windows, these exact techniques allow waiting at no penalty cost when a vehicle arrives earlier than the start time of service. This results in more waiting time as compared to the case when waiting is penalized[4)]. On the other hand, soft time windows are often encountered in practical freight transport. This could be desirable to develop exact solution approaches for the soft time windows variant as well. However, complex soft time windows constraints and time dependent costs have been the greatest barriers in the way of these efforts. Heuristics approaches such as insertion heuristics and genetic algorithms (GA) can efficiently handle complex soft time windows constraints, which can explain why heuristics (approximate) solutions are mostly used for the VRPSTW in city logistics-related research.

This paper presents a hybrid solution technique for the VRPSTW embedding a heuristics solution technique (Insertion heuristics) within the exact solution techniques (Column Generation) for the VRPHTW in order to improve the solution quality and to reduce computational times. The basic idea is to utilize dual information (shadow prices) obtained in the column generation scheme to improve the efficiency of the insertion heuristics. Reduced costs are obtained using these prices, and in a subproblem, the insertion heuristics searches for negative reduced cost routes (columns) for the master problem. The performance of the hybrid insertion heuristics has been evaluated, comparing its results on benchmark instances with a simple GA as well as with a similar hybridization approach developed by Qureshi *et al.*[5)] considering genetic algorithms as the heuristics subproblem.

2. Literature Review

A detailed overview of the VRPSTW and its solution techniques can be found in Taniguchi *et al.*[1)]. Many researchers have used heuristic techniques for the soft time windows environment with the idea to reduce the number of vehicles or total delivery cost[6)]. For example, Balakrishnan[7)] described the three simple heuristics

---

* Key words: logistics planning, Logistics, hybrid heuristics, soft time windows
** Student Member of JSCE, Dr. Eng., Post Doctoral Researcher, Faculty of Engineering, Kyoto University.
C-1 Kyotodaigaku Katsura, Nishikyo, Kyoto 615-8540, Tel. 075-383-3231, Fax. 075-950-3800.
*** Fellow Member of JSCE, Dr. Eng., Faculty of Engineering, Kyoto University.
C-1 Kyotodaigaku Katsura, Nishikyo, Kyoto 615-8540, Tel. 075-383-3229, Fax. 075-950-3800.
**** Full Member of JSCE, Dr. Eng., Faculty of Engineering, Kyoto University.
C-1 Kyotodaigaku Katsura, Nishikyo, Kyoto 615-8540, Tel. 075-383-3230, Fax. 075-950-3800.

for the VRPSTW based on the nearest neighbour, Clarke-Wright savings and space-time rules.

The Dantzig-Wolfe decomposition of the VRPTW results in the set partitioning master problem and an Elementary Shortest Path Problem with Resource Constraints (ESPPRC) as its subproblem[8]. However, many researchers have worked with various shortest path relaxations as subproblems to solve the VRPHTW[9), 10]. Instead of the NP-hard ESPPRC, this study employs an insertion heuristic in the subproblem (hereafter would be referred as IH-subproblem) to solve the Dantzig-Wolfe decomposition of the VRPSTW. Insertion heuristics have been used to solve many combinatorial optimization problems such as the flow-shop problems[11), 12] and the travelling salesman problem[13]. Insertion heuristics are also one of the earliest route-building heuristics originally developed for the VRPHTW[14]; therefore, in this study, a modified version is developed that can handle the soft time windows. Insertion heuristics have also been used in the initialization procedure for route improvement heuristics[15] and for metaheuristics[16] developed for the VRPTW.

In order to improve the efficiency of heuristics, so far the set partitioning linear optimization has been employed in many heuristics for the VRPSTW. For example, Calvete *et al.*[17] exploits 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. For the VRPHTW, Rochat and Taillard[18] used a heuristic approach, generating many candidate routes using intensified and diversified tabu search and then using set partitioning linear programming. Alvarenga *et al.*[16] used a specialized genetic algorithm to generate routes to be optimized with set partitioning formulation at the end of the algorithm.

All the above-cited references used the set partitioning linear programming after enumerating some or all possible candidate routes for the VRPTW, while this study utilizes the useful dual information, i.e., shadow prices, obtained every time when a set partitioning linear program is solved. The shadow prices are used to guide the optimization process in the IH-subproblem that in return provides routes with negative reduced cost to augment the set partitioning linear program. The set partitioning master problem and the IH-subproblem are solved in cycles. A similar approach has been adopted for a dynamic VRPHTW by Chen and Xu[19]. An excellent review of the heuristic methods applied to the VRPHTW and the VRPSTW can be found in Braysy and Gendreau[20), 21].

3. Model Formulation

The VRPSTW is defined on a directed graph $G = (V, A)$. The vertex set V includes the depot vertex 0 and the set of customers $C = \{1, 2, . . ., n\}$. The set K represents the set of identical vehicles with capacity q stationed at the depot. The arc set A consists of all feasible arcs (i, j), $i, j \in V$. A cost c_{ij} and a time t_{ij} is associated with each arc $(i, j) \in A$. The time t_{ij} includes the travel time on arc (i, j) and service time at vertex i, while a fixed vehicle utilization cost is added to all outgoing arcs from the depot. With every vertex $i \in V$ associates a demand d_i where $d_0 = 0$, and a time window $[a_i, b_i]$, which represents the earliest and the latest possible service start times.

This study incorporates the soft time windows constraint by extending the latest and earliest possible arrival time a_i to a_i' and b_i to b_i' as shown in Figure 1. The maximum penalty is considered equivalent to the cost of a dedicated single vehicle route only serving the concerned vertex. Taking c_l and c_e as the unit late arrival penalty cost and the unit early arrival penalty cost, respectively, a_i' and b_i' can be defined as Eq. (1) and Eq. (2).

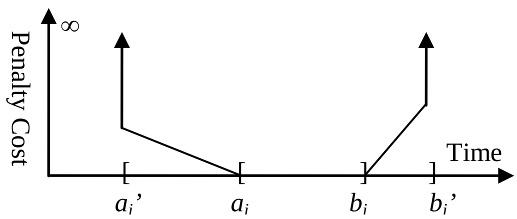


Figure 1 Penalty cost function for the VRPSTW

$$a_i' = \max\left[0, a_i - \frac{(c_{0i} + c_{i0})}{c_e}\right] \tag{1}$$

$$b_i' = \min\left[b_0 - t_{i0},\ \ b_i + \frac{(c_{0i} + c_{i0})}{c_l}\right] \tag{2}$$

Let s'_{jk} defines the service start time at a vertex $j \in C$ by a vehicle $k \in K$. For all arcs $(i, j) \in A$, the modified time dependent travel cost, c'_{ijk} is defined as a function of s'_{jk} (Eq. (3)). It may be noted that, even though arrival is allowed earlier than the start of time windows, a vehicle has to wait until a_i to start the service. This waiting time is considered as the early arrival penalty, as c_e is usually set equal to vehicle operating cost (VOC).

$$c'_{ijk} = \begin{cases} c_{ij}, \text{if } a_j \le s'_{jk} \le b_j \\ c_{ij} + c_l(s'_{jk} - b_j), \text{if } b_j < s'_{jk} \le b'_j \\ c_{ij} + c_e(a_j - s'_{jk}), \text{if } a'_j \le s'_{jk} < a_j \end{cases} \tag{3}$$

The VRPSTW can be mathematically formulated as:

$$\min \sum_{k \in K} \sum_{(i,j) \in A} c'_{ij} x_{ijk} \tag{4}$$

subject to

$$\sum_{k \in K} \sum_{j \in V} x_{ijk} = 1 \qquad \forall\, i \in C \tag{5}$$

$$\sum_{i \in C} d_i \sum_{j \in V} x_{ijk} \le q \qquad \forall\, k \in K \tag{6}$$

$$\sum_{j \in V} x_{0jk} = 1 \qquad \forall\, k \in K \tag{7}$$

$$\sum_{i \in V} x_{ihk} - \sum_{j \in V} x_{hjk} = 0 \qquad \forall\, h \in C, \qquad \forall\, k \in K \tag{8}$$

$$\sum_{i \in V} x_{i0k} = 1 \qquad \forall\, k \in K \tag{9}$$

$$a'_i \le s'_{ik} \le b'_i \qquad \forall\, i \in V, \qquad \forall\, k \in K \tag{10}$$

$$a_i \le s_{ik} \le b'_i \qquad \forall\, i \in V, \qquad \forall\, k \in K \tag{11}$$

$$s_{ik} + t_{ij} - s_{jk} \le (1 - x_{ijk}) M_{ijk} \qquad \forall\, (i, j) \in A, \qquad \forall\, k \in K \tag{12}$$

$$x_{ijk} \in \{0, 1\} \qquad \forall\, (i, j) \in A, \qquad \forall\, k \in K \tag{13}$$

The model contains two decision variables: s_{jk}' that determine the arrival time at a vertex $j \in C$ by a vehicle $k \in K$ as well as the travel cost of the arc (i, j) (Eq. (3)), and x_{ijk} that determine whether the arc (i, j) is used in the solution ($x_{ijk} = 1$) or not ($x_{ijk} = 0$). M_{ijk} is a large constant. Objective function (4) minimises the total delivery cost including the fixed vehicle utilization cost and the travel cost on the arcs as well as the penalty costs. Constraint (5) ensures that every customer must be serviced only once, while constraint (6) is the capacity constraint. Constraints (7), (8) and (9) are flow conservation constraints, defining that the route shall start and end at the depot, and if the vehicle travels to a customer node h, it must also travel from it. Constraint (10) specifies the relaxed time windows for the VRPSTW and restricts the arrival time at all vertices to be within their relaxed time windows [a_i', b_i']; whereas, constraint (11) restricts the service start time within [a_i, b_i']. Constraint (12) specifies that if a vehicle travels from i to j, service at j cannot start earlier than that at i. Finally, constraint (13) shows the integrality constraint.

4. Insertion Heuristics

(1) Hard Time Windows Case

In order to comprehend the differences between the insertion heuristics for the VRPSTW, a brief introduction of the classical insertion heuristics for the VRPHTW presented by Solomon[14)] is provided next. It is a sequential route-building heuristic where the route for one vehicle is completed before starting the route for another vehicle. To obtain multiple solutions, the routes are initiated based on various initialization criteria such as the first customer could be the one with smallest b_i value or which has the maximum c_{0j}. The remaining customers are added to this route using some insertion criteria based on a combination of obtained *savings* and/or change in the service start times. When no such new insertion is possible due to violation of either capacity or the time windows constraint, a new route is initialized. Let (i_0, i_1, i_2, . . ., i_m) be the current route with $i_0 = i_m = 0$. The service start time s_{i_r} and the waiting time w_{i_r} are known for $0 \le r \le m$. Insertion of a customer vertex u between i_{p-1} and i_p causes a *push forward* (Eq. (14)) in the schedule at the customer i_p that may change the values of s_{i_r} and w_{i_r} , $p \le r \le m$-1. Here, represents the push forward for the customer i_p, which is the difference between the previous (before insertion) service start time $s_{i_p}^{new}$ and the new (after insertion) service start time s_{i_p} of i_p. This push forward continues through to the end of the route (Eq. (15)) or until its value drops to zero after which the remaining part of the route is unaffected. The conditions $s_u \le b_u$ and $s_{i_r} + PF_{i_r} \le b_{i_r}$ provide the feasibility criteria as far as the time windows are concerned, for a feasible insertion position of the customer u in hard time windows case.

$$PF_{i_p} = s_{i_p}^{new} - s_{i_p} \ge 0 \tag{14}$$

$$PF_{i_{r+1}} = \max\{0,\ PF_{i_r} - w_{i_{r+1}}\},\ p \le r \le m-1 \tag{15}$$

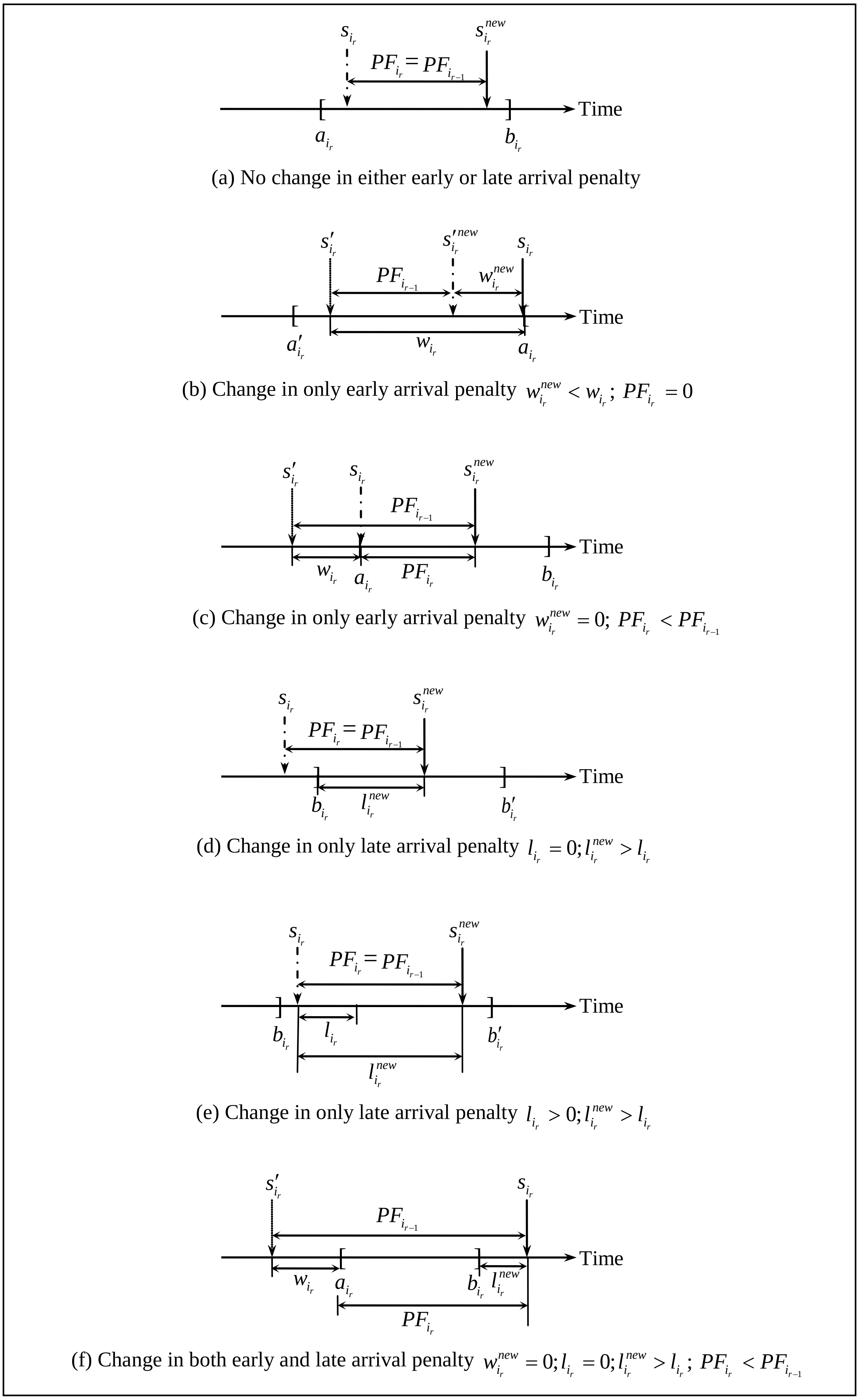
(2) Soft Time Windows Case

This study considers the VRPSTW, where the vehicle arrival is allowed within relaxed time windows [a_i', b_i'] and the arc cost depends on this arrival time (Eq. (3)). Therefore, in soft time windows (STW) case, for a partial route (i_0, i_1, i_2, . . ., i_m) with $i_0 = i_m = 0$, the late arrival time l_{i_r} must also be saved along with the service start time s_{i_r} and the waiting time w_{i_r} for $0 \le r \le m$-1. The push forward now can cause a change in the late arrival time l_{i_r} as well, along with the change in s_{i_r} and w_{i_r} , $p \le r \le m$-1. Furthermore, these changes not only affect the feasibility of the new route (as in the HTW case), these will also affect the insertion cost of the customer u in the STW case. Therefore, similar to Solomon[14)], the best feasible insertion place is determined using Eq. (16) for each unrouted customer u. However, an additional term is added to consider the changes in early and late arrival penalties for the customers i_r, $p+1 \le r \le m$-1 in the insertion cost criterion (Eq. (17)). It should be noted that the last term in Eq. (17) traces the changes in the early and late arrival penalties from the (p+1)th customer (the second customer after the inserted customer u) till the last customer on the route or until the point where push forward becomes zero. The first three terms of Eq. (17) also consider the arrival time dependent costs (as per Eq. (3)) as well. Finally, the best customer u^* to be inserted in the route, is obtained using Eq. (18). In the VRPSTW, feasibility conditions for time windows also change to $s_u \le b'_u$ and $s_{i_r} + PF_{i_r} \le b'_{i_r}$. Figure 2 shows some scenarios of the change in early and late arrival penalties resulting from the push forward. If a vehicle arrives earlier than a_{i_r} , the arrival time is marked with s'_{i_r} and the service start time by s_{i_r} . Similar to hard time windows (HTW) case, insertion of a customer vertex causes a push forward (Eq. (14)) that continues like a wave through to the end of the route (as shown in Figure 2(a), Figure 2(d) and Figure 2(e)), or its value is reduced along the route (as in the Figure 2(c) and Figure 2(f)) and eventually its value can drop to zero (Figure 2(b)).

$$c_1(i(u),u,j(u)) = \min[c_1(i_{p-1},u,i_p)],\ p = 1,\dots,m \tag{16}$$

$$c_1(i_{p-1},u,i_p) = c'_{i_{p-1},u} + c'_{u,i_p} - c'_{i_{p-1},i_p} + \sum_{r=p+1}^{m-1}(c_e(w_{i_r}^{new} - w_{i_r}) + c_l(l_{i_r}^{new} - l_{i_r})) \tag{17}$$

$$c_2(i(u^*),u^*,j(u^*)) = \min_u[c_1(i(u),u,j(u))] \tag{18}$$



(a) No change in either early or late arrival penalty

(b) Change in only early arrival penalty $w_{i_r}^{new} < w_{i_r}$; $PF_{i_r} = 0$

(c) Change in only early arrival penalty $w_{i_r}^{new} = 0$; $PF_{i_r} < PF_{i_{r-1}}$

(d) Change in only late arrival penalty $l_{i_r} = 0; l_{i_r}^{new} > l_{i_r}$

(e) Change in only late arrival penalty $l_{i_r} > 0; l_{i_r}^{new} > l_{i_r}$

(f) Change in both early and late arrival penalty $w_{i_r}^{new} = 0; l_{i_r} = 0; l_{i_r}^{new} > l_{i_r}$; $PF_{i_r} < PF_{i_{r-1}}$

Figure 2 Scenarios of the change in early and late arrival penalties incurred for push forward

5. Hybrid Insertion Heuristics (HIH)

In order to improve the quality of solutions produced by the insertion heuristics, these are subjected to some route-improving heuristics such as local search (for example see Savelsbergh[15]) as well as to metaheuristics (for example see Alverenga *et al.*[16]). To get the hybrid insertion heuristics (HIH), this study has embedded the insertion heuristics within the flexible framework of column generation scheme that can accommodate complex constraints and time dependent costs[22] such as in the VRPSTW. Column generation (Dantzig-Wolfe decomposition) decomposes the VRPSTW problem into a set partitioning master problem and an ESPPRC as its subproblem, which is a NP-hard problem itself. The role of the subproblem is to provide feasible shortest paths (single vehicle routes) subjected to constraints (Eq. (5)-Eq. (13)) to the set partitioning master problem (Eq. (19)- Eq. (21)) that selects a set of minimum cost routes (provided by the subproblem) with a constraint that every customer must be serviced once.

$$\min \sum_{p \in P} c_p y_p \tag{19}$$

subject to

$$\sum_{p \in P} a_{ip} y_p = 1 \qquad \forall i \in C \tag{20}$$

$$y_p \in \{0,1\} \qquad \forall p \in P \tag{21}$$

The set of all feasible paths is shown by P, where y_p takes a value of 1 if the path $p \in P$ is selected and 0 otherwise. The cost of the path p is denoted by c_p, and a_{ip} represents the number of times path p serves customer i. In the actual application, the set covering master problem is solved by replacing constraint (20) by (22), since the linear programming relaxation of set covering type master problem is more stable than the set partitioning type[9].

$$\sum_{p \in P} a_{ip} y_p \geq 1 \qquad \forall i \in C \tag{22}$$

This study uses the IH-subproblem instead of using the ESPPRC to generate the VRPSTW routes. The master problem is optimized using the linear programming, which also produces the shadow prices (dual variables' values). These prices (π_i) are used to find reduced costs (Eq. (23)) which guide the insertion heuristics to provide good quality routes with negative reduced costs used to augment the linear programming of the master problem. The master problem and the IH-subproblem are solved in cycles until the column generation procedure stops due to some stopping criteria. At this stage, if the VRPSTW solution contains fractional number of vehicles, routes with higher values of y_p (near to 1) are selected to formulate a partial solution *S'*, and a new smaller VRPSTW instance is created using remaining customers (i.e., *C/S'*). This process continues till an integer solution is found.

$$\overline{c'_{ij}} = c'_{ij} - \pi_i \qquad \forall\, i \in V \tag{23}$$

Usually the ESPPRC subproblem is solved for the Dantzig-Wolfe decomposition by removing the assignment constraint (4) and considering homogenous vehicles, thus finding routes for single vehicle only. The IH-subproblem is solved using the same constraint set as used in the VRPSTW formulation, which not only finds the shortest routes but assigns customers to the vehicles. The only difference is the use of the reduced cost $\overline{c'_{ij}}$ (as per Eq. (23)) in finding the insertion cost (Eq. (17)). Moreover, as a single run of subproblem, a stochastic version of the insertion heuristics (similar to the one used by Alverenga *et al.*[17]) is solved 50 times to generate many negative reduced costs routes from assorted insertion heuristics solutions. In the stochastic version, the first customer is chosen randomly while the remaining customers are inserted by minimizing the reduced cost of the route. The execution of the IH-subproblem stops prematurely when 100 negative reduced cost columns are found and all of these columns are added to the master LP problem.

6. Heuristics used for Comparison

To evaluate the performance of the Hybrid Insertion Heuristics (HIH) in terms of solution quality and computation time requirements, the results obtained using HIH are compared with the results from using a simple GA heuristics and another hybrid heuristics, namely, the Hybrid Genetic Algorithms with Column Generation Heuristics (HGACGH)[5]. The simple GA[5] uses a greedy look-ahead version of the insertion heuristics presented in the previous section in the population initialization. After selecting the first customer randomly, the route is extended by adding customers based on minimum insertion cost next to the first customer and so on. An ordered-based two-point crossover is used at a rate of 98% to maintain the structure of the chromosome, which represents a complete feasible solution of the VRPSTW instance in the form of a chain of customers. Simple swap mutation at a mutation rate of 5% is used to perturb the genetic search. The optimization effort is undertaken for larger problems by using the number of generations as a function of the number of customer vertices (Number of generations = 250 x Number of Customers). Furthermore, the population is regenerated after every 500 generations. During this step, a new population is generated by keeping 4% of the current population and the remaining 96% is generated as that in the initialization.

A genetic algorithm is used to solve the subproblem (hereafter would be referred as GA-subproblem) in HGACGH[5]; it uses the same initialization and crossover as those used in the simple GA. However, the chromosomes are structured to include the information on number of routes so that a chromosome with a single customer route can be distinguished. The Single Customer Route Elimination Mutation (SCREM) was then used to insert that customer (i.e. target customer) at a feasible place in other routes present in the same chromosome. At every column generation iteration in HGACGH, the GA-subproblem is solved with 30 generations to get many columns with negative reduced costs.

7. Results and Discussion

The R1-type instances in Solomon's benchmark instances were used to evaluate the performance of the HIH as compared to that of the simple heuristics and the HGACGH. Customers are located randomly in these instances (Figure 3), and each instance contains 100 customers, where smaller instances can be generated by taking the first few customers. R101, R102 and R103 instances with 100 and 50 customers were used; these instances vary in the number of customers with binding time windows (i.e. time windows having the widths of 10 units). All customers in R101 have the binding time windows, whereas R102 and R103 have 75% and 50% of the customers with the binding time windows, respectively. A suffix of STW is added to the instance name to indicate that soft time windows are considered. For example, R101-50-STW represents an instance derived from R101-100 Solomon's benchmark instance, with 50 customers and soft time windows.

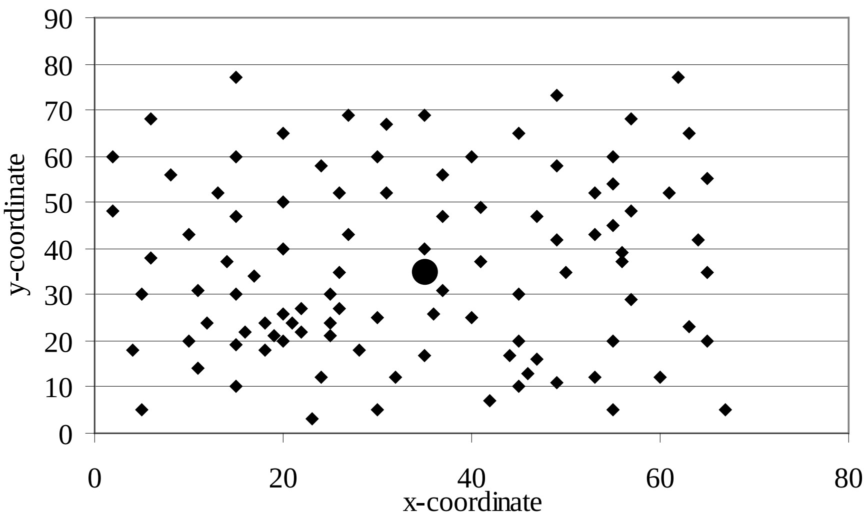


Figure 3 Customer locations in R1-type Solomon's benchmark instance

The algorithms were coded and implemented in MATLAB on a PC with 2.41 GHz AMD Athlon with 64 x 2 dual core processors and 2 GB of RAM. The maximum number of column generation iterations was set to 75 runs for the hybrid heuristics for the first part (before problem reduction step) and 50 runs for the reduced problems (after problem reduction step). A vehicle operation cost (VOC) of 14.02 JPY/minute was taken; while the fixed cost for a vehicle was set to 10417.5 JPY. The unit early arrival penalty cost was assumed equal to the VOC, whereas the unit late arrival penalty was 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.*[1)]; Yamada *et al.*[2)]; Taniguchi and Thompson[6)]).

Figure 2 shows the total delivery cost and computation time for all the three heuristics, i.e., the HIH, HGACGH and simple GA. The term total delivery cost is used here to indicate the total cost of a VRPSTW solution that includes the fixed vehicle cost, travelling cost, early and late penalties for all routes in that particular solution. The average computation times in 100 and 50 customers' instances in HIH were 86.7% and 94.5% less than those in the simple GA, respectively; whereas the corresponding figures in the HGACGH were 73% and 80.5%, respectively. On the total delivery cost side, the results do not show a general trend but overall the HIH produced better solutions with 5.4% and 8% average cost reductions as compared to simple GA in 50 and 100 customers' instances. The average cost reductions in HGACGH as compared to simple GA were 4.7% and 2.3% in 50 and 100 customers' instances, respectively. As it is observed, in terms of the computation time, the simple GA is outperformed by both hybrid approaches, and therefore the comparison between HIH and HGACGH is only focused hereafter. Tables 1 and 2 show the detailed cost components and some algorithmic details for these hybrid heuristics.

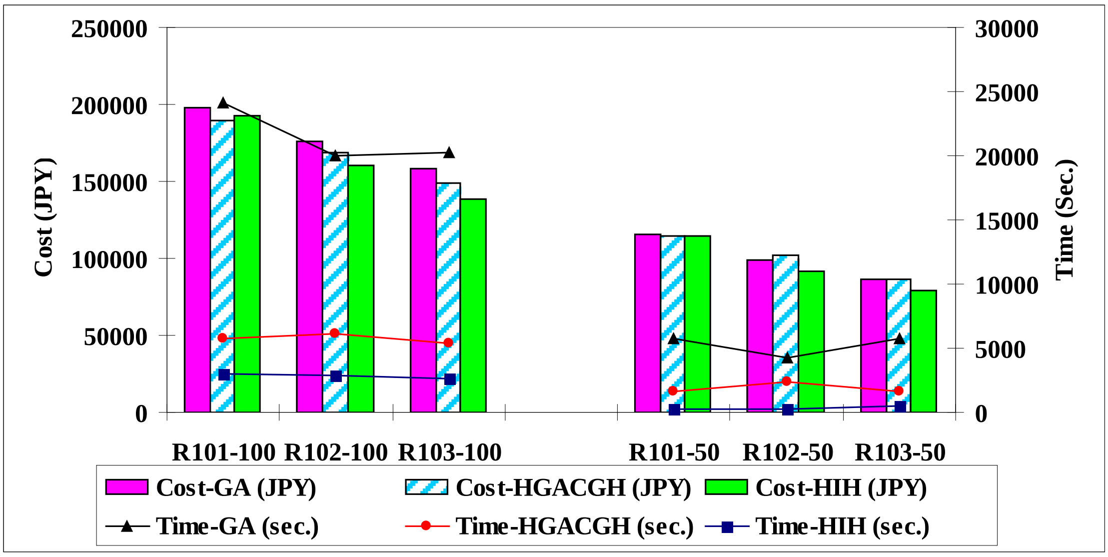


Figure 4 Comparison between HIH, HGACGH and simple GA

The best results found using the HIH and HGACGH on every test instance are shown in Table 1. The total delivery cost is divided into various components. Operation cost represents the total delivery time composed of vehicles' running time and the service times at customers. Fixed cost relates closely to the number of vehicles required. Therefore, the difference between the fixed costs in the HIH and the HGACGH shows that the HIH could produce solutions with less number of vehicles. It seems that the HIH is very efficient in reducing the number of vehicles required; however, the amount of late arrival penalties has also increased considerably as compared to the solutions obtained using the HGACGH. This result shows the traditional trade-off between the number of vehicles and the late arrivals. On the other hand, the amount of early arrival penalties emphasize that the solutions obtained in the HIH contain less waiting time, which can also help reduce on-street parking and traffic congestion.

Table 2 contains the average algorithmic results of the HIH and HGACGH, showing the total number of column generation iterations, total number of columns (negative cost routes) added to the master problem LP and the total computation time in seconds. It illustrates that the convergence in the HIH was relatively fast as compared to the HGACGH. This resulted in efficient performance of the HIH in terms of computation time, averagely saving 83.5% and 50.4% in computation time in 50 and 100 customers' instances, respectively as compared to the HGACGH. Performance of the HIH, in context of total delivery cost (Table 1), was also satisfactory, with average cost reductions of 3.5% and 6% being observed for 50 and 100 customers' instances. However, in R101-50, the HIH provided more cost and an increase of 0.7% in total delivery cost was observed as compared to HGACGH, and for R101-100 the difference increased to 1.7%.

Table 1 Cost comparisons between HIH and HGACGH (in JPY)

Instance	HIH					HGACGH				
	Operating Cost	Late Arrival Penalty	Early Arrival Penalty	Fixed Cost of Vehicles	Total Delivery Cost	Operating Cost	Late Arrival Penalty	Early Arrival Penalty	Fixed Cost of Vehicles	Total Delivery Cost
R101-50-STW	14506.5	16781.9	330.872	83340	114959	15354.7	4002.71	999.626	93757.5	114115
R102-50-STW	13567.2	4864.94	298.626	72922.5	91653.2	14777.1	3792.41	691.186	83340	102601
R103-50-STW	11809	5173.38	148.612	62505	79636	11724.9	1345.92	629.498	72922.5	86622.8
R101-100-STW	23511.5	33234.4	506.122	135428	192680	23311.1	8208.71	1609.5	156263	189392
R102-100-STW	22040.8	12526.9	382.746	125010	159960	22513.3	9274.23	1473.5	135428	168689
R103-100-STW	19375.6	4262.08	406.58	114593	138637	19905.6	3105.43	1016.45	125010	149037

Table 2 Algorithmic details of HIH and HGACGH results for the test instances

Instance	HIH			HGACGH		
	Iterations	Columns	Time (sec.)	Iterations	Columns	Time (sec.)
R101-50-STW	37	1187	217.10	57	1072	1220.06
R102-50-STW	43	1472	249.31	86	1225	1987.00
R103-50-STW	48	1575	286.01	77	1266	1605.90
R101-100-STW	118	5046	2959.03	140	2570	5784.61
R102-100-STW	125	5792	2909.84	151	2693	5847.62
R103-100-STW	113	3628	2512.26	155	2725	5850.82

8. Conclusion

This study presented a hybrid approach for solving the Vehicle Routing and scheduling Problem with Soft Time Windows (VRPSTW) combining the column generation and a modified insertion heuristics to deal with the soft time windows and penalty costs. The dual information obtained in the column generation master problem was used to guide the optimization in the hybrid insertion heuristics (HIH). This resulted in rapid convergence to a good solution. As a result, the hybrid insertion heuristics (HIH) performed much better than a simple genetic algorithms both in terms of total delivery cost and computation time. As compared to another hybrid scheme using the genetic algorithm in the subproblem (HGACGH)[5], the HIH was able to considerably reduce the computation time. Computation results showed that the HIH was very efficient in reducing the number of vehicles but it resulted in higher late arrival penalties; however, the total delivery cost was also less than the HGACGH in most of the cases. Thus, the HIH provide logistics managers and the policy planers of city logistics-related schemes, a faster and better solution method for the VRPSTW as compared to the HGACGH, to evaluate their policies swiftly and confidently.

References:

1) Taniguchi, E., Thompson, R.G., Yamada, T. and Van Duin J.H.R.: City Logistics; Network Modeling and Intelligent Transport Systems, Pergamon, Oxford, 2001.
2) Qureshi, A.G., and Hanaoka, S.: Analysis of the effects of cooperative delivery system in Bangkok, in Recent Advances in City Logistics proceedings of the 4th International Conference on City Logistics, Langkawi, Malaysia, 2005, Taniguchi, E. and Thompson, R.G. eds., Elsevier, Oxford, pp. 293-306, 2005.
3) Yamada, T., Taniguchi, E. and Itoh, Y.: Co-operative vehicle routing model with optimal location of logistics terminals, in City Logistics II, Taniguchi, E., and Thompson, R.G. eds., Institute for City Logistics, Japan, pp. 139-153, 2005.
4) Qureshi, A.G., Taniguchi, E. and Yamada, T.: Effects of relaxing time windows on vehicle routing and scheduling**,** Infrastructure Planning Review, Vol. 24, pp. 927-936, 2007.
5) Qureshi A.G., Taniguchi, E. and Yamada, T.: A hybrid genetic algorithm for VRPSTW using column generation, to be appeared in Innovations in City Logistics, Taniguchi, E., and Thompson, R.G., eds., Nova Publishers, New York, 2008.
6) Taniguchi, E., and Thompson, R.G.: Modeling city logistics, Transportation Research Records, Vol. 1790, pp. 45-51, 2002.
7) Balakrishnan, N.: Simple heuristics for the vehicle routing problem with soft time windows, Journal of the Operational Research Society, Vol. 44, pp. 279-287, 1993.
8) Feillet, D., Dejax, P., Gendreau, M. and Gueguen, C.: An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems, Networks, pp. 216-229, 2004.
9) Desrochers, M., Desrosiers, J. and Solomon, M.: A new optimization algorithm for the vehicle routing problem with time windows, Operations Research, Vol. 40(2), pp. 342-354, 1992.
10) Irnich, S., and Desaulniers, G.: Shortest path problems with resource constraints, in Column Generation, G. Desaulniers, Desrosiers, J., & Solomon M.M., eds., Springer, New York, pp. 33-65, 2005.
11) Rajendran, C.: Heuristic algorithm for scheduling in a flowshop to minimize total flowtime, International Journal of Production Economics, Vol. 29, pp. 65-73, 1993.
12) Kurz, M.E., and Askin, R.G.: Scheduling flexible flow lines with sequence-dependent setup times, European Journal of Operational Research, Vol. 159, pp. 66–82, 2004.
13) Mosheiov, G.: The travelling salesman problem with pick-up and delivery, European Journal of Operational Research, Vol. 79, pp. 299-310, 1994.
14) Solomon, M.M.: Algorithms for the vehicle routing and scheduling problem with time windows constraints, Operations Research, Vol. 35(2), pp. 254-265, 1987.
15) Savelsbergh, M.W.P.: The vehicle routing problem with time windows: Minimizing route duration, ORSA Journal on Computing, Vol. 4, 146-154, 1992.
16) Alvarenga, G.B., Mateus, G.R. and Tomi, G.de.: A genetic and set partitioning two-phased approach for the vehicle routing problem with time windows, Computers & Operations Research, Vol. 34, pp. 1561-1584, 2007.
17) Rochat Y., and Taillard, E.D.: Probabilistic diversification and intensification in local search for vehicle routing, Journal of Heuristics, Vol. 1, pp. 147-167, 1995.

18) Calvete, H.I., Gale, C., Oliveros M.J. and Valverde, B.S.: A goal programming approach to vehicle routing problem with soft time windows, European Journal of Operational Research, Vol. 177, pp. 1720-1733, 2007.
19) Chen, Z., and Xu, H.: Dynamic column generation for dynamic vehicle routing with time windows, Transportation Science, Vol. 40, 74-88, 2006.
20) Braysy, O., and Gendreau, M.: Vehicle routing problem with time windows, Part I: Route construction and local search algorithms, Transportation Science, Vol. 39, pp. 104-118, 2005a.
21) Braysy, O., and Gendreau, M.: Vehicle routing problem with time windows, Part II: Metaheuristics, Transportation Science, Vol. 39, pp. 119-139, 2005b.
22) Fisher, M.L.: Vehicle Routing, in Network Routing: Handbooks in Operations Research and Management Science Vol. 8, Ball, M.O., Magnanti, T.L., Monma, C.L., and Nemhauser, G.L., eds., North-Holland, Amsterdam, pp. 1-33, 1995.

HYBRID INSERTION HEURISTICS FOR VEHICLE ROUTING PROBLEM WITH SOFT TIME WINDOWS

Ali Gul Qureshi **, Eiichi Taniguchi*** and Tadashi Yamada****

This paper presents a hybrid insertion heuristics (HIH) for the Vehicle Routing and scheduling Problem with Soft Time Windows (VRPSTW) that can be used as an analysis tool to evaluate the effectiveness of many infrastructure planning and management measures belonging to City Logistics such as cooperative delivery systems and ideal location of logistics terminals. The dual information obtained in the column generation master problem was used to guide the optimization in the HIH. Consequently the objective function rapidly converged to a good solution, which resulted in better performance as compared to a simple genetic algorithms heuristics and to another hybrid heuristics.