considering time windows or due date constraints. The existing literature on (VRSPTW) has been directed primarily at special structures such as the dial-a-ride problem and the school bus scheduling problem. In these areas some progress has been made. Initial developments can also be seen for the (TSPTW). Nevertheless, capacitated problems with time windows have yet to be analyzed. Turning now to the analysis of algorithms, limited computational testing has been the sole means utilized in assessing performance. To our knowledge, with the exception of two papers for vehicle routing, no attempts have been made to obtain analytical results about the behavior of routing and scheduling algorithms. In closing we should remark that research on (VRSPTW) is just beginning. This is witnessed by the majority of papers still in working paper form.

CHAPTER 3

THE VEHICLE ROUTING AND SCHEDULING PROBLEM WITH TIME WINDOW CONSTRAINTS; FORMULATION AND HEURISTICS

3.1 Introduction

In this chapter we introduce a mixed integer programming formulation for the (VRSPTW) and describe various approximate methods for its solution. To begin with, let us define a time window for a customer, say i, by giving an earliest (e_i) and latest (d_i) service time.

We distinguish several situations with respect to the constraints imposed on the time of delivery to customers:

i) $e_i = 0$, $d_i = T$, for all i

ii) $e_i = d_i = t_i$, for all i

iii) $e_i = 0$, $d_i < T$, for some i

iv) $e_i > 0$, $d_i = T$ for some i

v) $e_i \geq 0$, $d_i \leq T$, for some i, where T is a scalar chosen to be larger than the total schedule time of any feasible route, and t_i is the arrival time at customer i. Case i) is the vehicle routing problem, while case ii) is the vehicle scheduling problem. In cases iii) and iv), one-sided time windows are present in the form of due dates (case iii) or earliest delivery times (case iv). Such time-windows are encountered, for example, in some dial-a-ride problems where customers specify desired pick-up times or latest delivery times. Case v) is the general (or "mixed") problem with time windows, where two-sided time windows are imposed on possible service times. When time windows are present, both the routing and scheduling of vehicles needs to be performed.

We now introduce the (VRSPTW) formally. Our formulation is based on a model for the basic routing problem given by Fisher and Jaikumar(1981). The following notation is needed:

<u>Constants</u>:

- K = number of vehicles.
- n = number of demand points; 0 denotes the central depot.
- b_k = capacity of vehicle k.
- q_i = demand of customer i.
- c_{ijk} = cost of direct travel for vehicle k, from customer i to customer j.

t_{ij} = direct travel time from customer i to customer j.

s_i = service time at customer i.

e_i = earliest time allowed for delivery to customer i.

d_i = latest time for delivery to customer i.

T = a scalar chosen to be larger than the travel time of any feasible route.

<u>Variables</u>:

$$y_{ik} = \begin{cases} 1, & \text{if customer i's order is delivered by vehicle k.} \\ 0, & \text{otherwise.} \end{cases}$$

$$x_{ijk} = \begin{cases} 1, & \text{if vehicle k travels directly from customer i to customer j.} \\ 0, & \text{otherwise.} \end{cases}$$

t_i = arrival time at customer i.

s = start $t_{0,s}^{k}$ = departure time of vehicle k from the depot.

f = finish $t_{0,f}^{k}$ = arrival time of vehicle k at the depot.

We now state an MIP formulation for the problem of routing and scheduling capacitated vehicles subject to time windows constraints.

MIP Formulation for the Vehicle Routing and Scheduling Problem with Time Windows (VRSPTW).

min $\quad \Sigma_{ijk} \; c_{ijk}x_{ijk}$, subject to: $\qquad$ (3.1)

s.t.

$$\Sigma_i \; q_i y_{ik} \le b_k, \; k=1,\ldots,K \qquad (3.2)$$

$$\Sigma_k \; y_{ik} = \begin{cases} K, & i=0 \\ 1, & i=1,\ldots,n \end{cases} \qquad (3.3)$$

$$\Sigma_i \; x_{ijk} = y_{jk}, \; j=0,\ldots,n; \; k=1,\ldots,K \qquad (3.4)$$

$$\Sigma_j \; x_{ijk} = y_{ik}, \; i=0,\ldots,n; \; k=1,\ldots,K \qquad (3.5)$$

$$t_j \ge t_i + s_i + t_{ij} - (1-x_{ijk})T, \; \begin{matrix} i,j=1,\ldots,n \\ k=1,\ldots,K \end{matrix} \qquad (3.6)$$

$$t_{0,f}^{k} \ge t_i + s_i + t_{i0} - (1-x_{i0k})T, \; \begin{matrix} i=1,\ldots,n \\ k=1,\ldots,K \end{matrix} \qquad (3.7)$$

$$t_j \ge t_{0,s}^{k} + s_i + t_{0j} - (1-x_{0jk})T, \; \begin{matrix} j=1,\ldots,n \\ k=1,\ldots,K \end{matrix} \qquad (3.8)$$

$$e_i \le t_i \le d_i \quad , \; i=1,\ldots,n \qquad (3.9)$$

$$e_0 \le t_{0,p}^{k} \le d_0 \; , \; p=s,f, \; k=1,\ldots,K \qquad (3.10)$$

$$y_{ik} = 0 \text{ or } 1, \; i=0,\ldots,n; \; k=1,\ldots,K \qquad (3.11)$$

$$x_{ijk} = 0 \text{ or } 1, \; i,j=0,\ldots,n; \; k=1,\ldots,K \qquad (3.12)$$

$$t_i \ge 0, \; i=0,\ldots,n \qquad (3.13)$$

Constraints (3.2) ensure that each route's demand is within the capacity limit of the vehicle servicing that route. (3.3) state that each route originates and terminates at the depot, and that every customer is on exactly one route. For every k, and y_{ik} fixed to satisfy (3.2)-(3.3) and (3.11), constraints (3.4)-(3.10), (3.12) are imposed on the customers assigned to vehicle k. (3.4)-(3.5) constrain a route to enter and leave each customer location exactly once. Constraints (3.6)-(3.8) ensure compatible arrival times. Moreover, (3.6) eliminate all oriented subtours. A proof of their validity is given below. (3.9) restrict the arrival time at a customer to its time window. (3.10) are depot time window constraints on vehicle departure and arrival times. We will show now that constraints (3.6) eliminate all the oriented subtours. For this, let $T(y_{ik}) = \{ i : y_{ik} = 1 \}$ and $T'(y_{ik}) = T - \{0\}$.

<u>Lemma 3.1</u>. For given k and y_{ik} fixed to satisfy (3.2)-(3.3) and (3.11), the constraints:

$$t_j \geq t_i + t_{ij} + s_i - (1 - x_{ijk})T, \quad i,j \in T'(y_{ik})$$

are satisfied if and only if there are no oriented subtours.

Proof: Clearly these constraints are satisfied when there are no oriented subtours. We prove now the second half of the lemma by contradiction. Let $T'(y_k) = \{i_1,\ldots,i_{n(k)}\}$, and let the variables $\{ x_{ijk} : i,j=i_l, l=1,\ldots,n(k), i\neq j\}$ and $\{ t_i : i=i_l, l=1,\ldots,n(k)\}$ satisfy the above constraints. Assume now that there exists an oriented subtour of order, say p. Without loss of generality, assume that the subtour is $\{i_1,\ldots,i_p\}$. We must have:

$$x_{i_1 i_2 k} = x_{i_2 i_3 k} = \ldots = x_{i_p i_1 k} = 1.$$

Since

$$t_{i_m} \geq t_{i_{m-1}} + s_{i_{m-1}} + t_{i_{m-1} i_m}, \; m=2,\ldots,p,$$

trivially,

$$t_{i_p} \geq t_{i_1} + \Sigma_{m=2}^{p} t_{i_{m-1} i_m} \tag{3.14}$$

Also,

$$t_{i_1} \geq t_{i_p} + s_p + t_{i_p i_1} \tag{3.15}$$

From (3.14) and (3.15) it follows that $t_{i_{m-1} i_m} = t_{i_p i_1} = 0$ and $s_m = 0$ for all m. We have reached a contradiction; therefore there cannot be any oriented subtours.

It follows from the above lemma that the usual subtour elimination constraints, found in (VRP) formulations are superfluous in our formulation of the (VRSPTW). Concerning the scalar T used in constraints (3.6), note that it is sufficient to take $T=\max\{d_i+s_i+t_{ij}-e_j\}$. Alternatively, for each arc (i,j), we can define its associated constant T as the quantity in the above parantheses.

Miller et al.(1960) have introduced the following subtour elimination constraints for the (TSP):

$$a_i - a_j + (n+1)\, x_{ij} \leq n \qquad (3.16)$$
$$i=0,1,\ldots,n;\ j=1,\ldots,n+1,\ i\neq j$$

where the home city is labelled as city 0 and city n+1, and a_i is a real number associated with city i, $1\leq a_i\leq n+1$, $i=1,\ldots,n+1$, and $x_{ij}=x_{ijk}$ with k=1. If we let $t_i=a_i$, $t_{ij}=x_{ij}$, T=n, $s_i=0$, all i,j, then one can see that (3.16) is a particular case of the constraints (3.6). In other words, in the classical (TSP), where only the sequencing and not the scheduling of the customers is required, one can assume unit travel times, which leads to the constraints (3.16). This generalization of the classical (TSP) subtour elimination constraints has been derived independently by Desrosiers et al.(1983A), who use it in the context of school bus scheduling.

A solution for the (VRSPTW) provides an allocation of customers to vehicles (variables y_{ik}), the routes (variables x_{ijk}), i.e. the sequence in which the customers are going to be visited, and the vehicle schedules (variables t_i). We assume that if a vehicle arrives early at a customer location, it will wait, i.e. $t_j=\max\ \{e_j, \Sigma_{ik}\ (t_{ij}+s_i+t_i)x_{ijk}\}$.

The above formulation is very flexible, and, as such, it can accomodate a number of important problem complexities. To start with, we should remark that in the vehicle routing field, one usually assumes a fleet of identical vehicles i.e. $b_k=b$, for all k; hence $c_{ijk}=c_{ij}$, for all k. The fleet size is assumed to be large enough to permit the existence of a feasible solution. Moreover, the cost of travel c_{ij} is considered to be a function of distance (usually $c_{ij}=d_{ij}$, where d_{ij} is the distance between i and j) and sometimes of travel time ($c_{ij}=t_{ij}$). A more general criterion is provided by the weighted combination of the above criteria, i.e. $c_{ij}=a_1d_{ij}+a_2t_{ij}$.

In the presence of time windows the primary objective becomes the minimization of the total routing and scheduling costs, which include not only the total travel distance and time costs considered for routing problems, but also the cost of waiting time which is inccured when a vehicle arrives too early at a customer location or when the vehicle is loaded or unloaded. For example, in many realistic environments, the unit cost of travel involves a cost per unit distance (ex: gasoline) and a cost per unit time (ex: driver's wages). Then, $c_{ij} = a_1d_{ij} + a_2(t_j-t_i)$, $a_1+a_2=1$.

This objective can be expressed alternatively as Σ_{ijk} $a_1 d_{ij} x_{ijk} + \Sigma_k\ a_2(t_{0,f}^{\,k} - t_{0,s}^{\,k})$. Even in situations where the cost per unit time is sunk (ex: driver paid a salary), the time cost must be considered. Waiting time contributes to the elongation of total schedule time which can lead to overtime periods. One way to express this contingency in our formulation is to let $c_{ij}=a_1 d_{ij} + a_2[(t_j-T)^+ - (t_i-T)^+]$. One can restrict or even eliminate overtime periods by introducing length of route-time constraints. Such constraints on the length of the scheduling horizon arise frequently in practice from legal or contractual conditions. They can be handled very easily in the above formulation by letting the depot due date d_0 equal the desired route-time and $d_i=d_0-t_{i0}-s_i$.

The above discussion underscores the importance of the waiting time costs in routing and scheduling problems with time windows. In certain cases, the minimization of total waiting time is even the primary objective (see Pullen and Webb(1967)). To minimize this criterion, one can define the waiting time at j as $w_j=[e_j - \Sigma_{ik}(t_{ij}+s_i+t_i)\ x_{ijk}]^+$ and let $c_{ijk}=(e_j-t_i-s_i-t_{ij})^+$.

Our model has several other important features. For example, it is not always the case that $e_0=0$ and all vehicles leave the depot at time zero, i.e. $t_{0,s}^{\,k}=0$, for all k. In some situations, vehicles can be used for additional trips. By defining the model variables for routes rather than for vehicles, a solution to the model will provide the starting and ending time of each route, and, hence, a subsequent assignment of vehicles to routes can be made. Different allowable working periods for the vehicles can be implemented through the use of multiple time windows on the depot $e_0^k \leq t_{0,p}^{\ k} \leq d_0^k$, $p=s,f$. Multiple time windows can also be easily incorporated in the model.

As mentioned above, vehicle routing and scheduling problems involve a predetermined number of vehicles, say K, all of the same capacity. It is therefore assumed that the fixed vehicle costs are sunk, i.e. the fleet of vehicles has already been aquired, and only variable routing and scheduling costs need to be considered. This tactical or operational problem becomes a concern only after the strategic problem of how many vehicles to purchase or lease is solved. In the latter case, both fixed vehicle and variable routing and scheduling costs need to be considered. Our model can accomodate this fleet size problem. To see this, let $f_k=f$, for all k, be the fixed vehicle cost and $c'_{0jk}= f_k+c_{0jk}$. The number of vehicles, K, now becomes a decision variable, and one needs to impose the additional constraint $K^* \leq K \leq n$, where K^* is a lower bound on the number of vehicles used. One possible value for K^* is given by $\sum_{k=1}^{K^*-1} b_k < \sum_{i=1}^{n} a_i \leq \sum_{k=1}^{K^*} b_k$.

If f_k is sufficiently large, it is never worthwhile to increase the number of vehicles in order to reduce the variable routing and scheduling cost. In such cases, the model will first minimize the number of vehicles and for this number, it will minimize the variable cost. Otherwise there exist tradeoffs between fleet size and operating costs. A generalization of the vehicle fleet size problem is the fleet size and mix problem, where vehicles of different costs and capacities have to be considered. The solution to this problem would then precede that of the routing and scheduling problem if this latter problem considers a non-homogeneous fleet.

Considering now the computational complexity of problems with time windows, since the (VRP) is NP-hard (see Lenstra and Rinnooy Kan(1981)), by reduction, the (VRSPTW) is NP-hard. As such, it is very improbable that an optimal solution could be found in polynomial time. Therefore, the development of heuristic algorithms for this problem class is of primary interest. We focus on this topic in the following section.

3.2 Heuristics for (VRSPTW)

We now decribe approximate methods for the solution of (VRSPTW) formulated in the preceding section. Of primary importance to the effectiveness and efficiency of such methods is the way in which the time window constraints are incorporated in the solution process. Since we will concentrate on route building procedures, let us first examine the necessary and sufficient conditions for time feasibility when inserting a customer, say l, as the first customer on a partially constructed route. Let (0,0), (f_1,t_{f_1}), $(f_2,t_{f_2}),\ldots,$ (f_m,t_{f_m}), $(0,t_0)$ be the schedule for the partial route currently having $f=f_1$ as the first customer. Denote by S(i) the set of successors of i on its route. An immediate necessary condition is that the new arrival time at f_1 be less than the latest time for delivery at f_1, i.e.

$$t^n_{f_1} = t_l + s_l + t_{lf_1} \leq d_{f_1} \qquad (3.17)$$

If $t^n_{f_1} > e_{f_1}$, then the arrival times at all customers in $S(f_1)$ could potentially be pushed forward in time, possibly making some of them infeasible. Define the <u>push forward</u> in the schedule at f_1 as:

$$PF_{f_1} = t^n_{f_1} - t_{f_1}$$

If $t_{f_2} > e_{f_2}$, then $PF_{f_2} = PF_{f_1}$, and hence the new arrival time at f_2 is:

$$t^n_{f_2} = t_{f_2} + PF_{f_2}$$

In this case, the push forward is fully propagated at f_2 (see the diagram below).

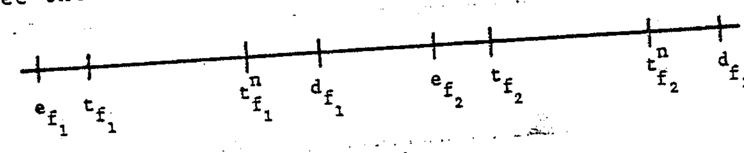


Diagram 3.1

We must require that:

$$t^n_{f_2} \leq d_{f_2}.$$

If $t_{f_2} \leq e_{f_2}$ and $w_{f_2} < PF_{f_1}$ (where w_i is the waiting time at i) ,then $PF_{f_2} = PF_{f_1} - w_{f_2}$, i.e. some of the push forward is absorbed by the waiting time at f_2. We must have that:

$$t^n_{f_2} = t_{f_2} + PF_{f_2} \leq d_{f_2}.$$

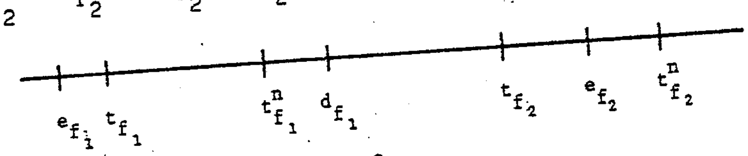
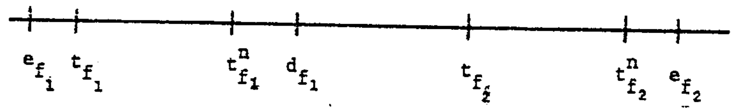


Diagram 3.2

If $t_{f_2} \leq e_{f_2}$ and $w_{f_2} \geq PF_{f_1}$,then obviously $PF_{f_2} = 0$, and therefore, all the customers in $S(f_2)$ will remain feasible. In other words, the push forward in the schedule is completely absorbed at f_2 (see diagram 3.3, below).



<u>Diagram 3.3</u>

It is clear by a simple induction argument that the process of sequentially examining the customers in $S(f_1)$ for time feasibility continues until we find some customer, say f_p, $p < m$, such that $PF_{f_p} = 0$, or the customer is time infeasible, or, in the worst case, all the customers in $S(f_1)$ are examined. We have just proved:

<u>Lemma 3.2</u>. The necessary and sufficient conditions for a link (1,f) to be time feasible are that:

$t^n_{f_1} \leq d_{f_1}$, and

$$t_{f_i} + PF_{f_i} \leq d_{f_i}, \tag{3.18}$$

for all $f_i \varepsilon S(f_1)$ such that $PF_{f_i} > 0$, $1 \leq i \leq m$, where

$$PF_{f_i} = \begin{cases} PF_{f_{i-1}}, \text{ if } t_{f_i} > e_{f_i} \\ PF_{f_{i-1}} - w_{f_i}, \text{ if } t_{f_i} \leq e_{f_i} \text{ and } w_{f_i} < PF_{f_{i-1}}. \end{cases}$$

Let us now describe several heuristic methods for the solution of (VRSPTW).

3.2.1 Savings Heuristics

An approach to solving (VRSPTW) is to extend the savings heuristic of Clarke and Wright(1964), originally proposed for the (VRP). This is a tour building heuristic. Such heuristics are characterized by the sequential addition of a link between two customers, guided by some measure of cost savings, until all customers have been assigned to some route. For the savings method this measure is:

$$s_{ij} = c_{i0} + c_{0j} - \Theta c_{ij}$$

The route shape parameter Θ was independently introduced by Gaskell(1967) and Yellow(1970). At every step in the heuristic process, the time and capacity constraints are checked for violation. Whereas, in routing, the direction of travel on a tour is unimportant, we now have to account for orientation, due to the existence of time windows. Two routes with end customers i and j, respectively, have compatible orientations if i is first (last), and j is last (first), as the following diagrams illustrate:

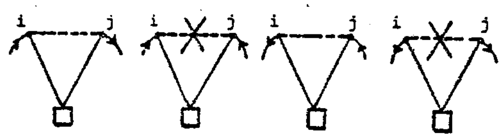


Diagram 3.4 Compatible orientations.

Hence, the admissible links are from the last customer(l) on a route to the first customer(f) on another. We can now describe the extension of the savings approach to the (VRPSTW). Two distinct subtours containing customer i and j, respectively, can be joined if:

a) i and j are not in the same subtour;

b) neither i nor j is an interior point in a subtour;

c) the combined subtour containing both i and j has a total demand not exceeding vehicle capacity;

d) The subtours containing i and j, respectively, have compatible orientations, and the link (i,j) is time feasible.

Our computer implementation uses lemma 3.2 to efficiently test time feasibility when linking two customers. In our computational tests, we have found our time feasibility conditions to be clearly superior to the explicit testing of time feasibility at each customer. We have also used list processing and heapsort structures, as proposed by Golden et al.(1977) for the Clarke and Wright heuristic, to reduce the core requirements and computation time. For (VRSPTW), a choice exists as to how to orient the initial partial routes. After preliminary computational experience, we have decided to select the customer with the earlier due date as the first customer on its route.

As presented, the savings heuristic could find it profitable to join two customers very close in distance but far apart in time. Such links lead to the introduction of extended periods of waiting time. As discussed before, waiting time can have a high opportunity cost as far as total schedule cost and the number of vehicles are concerned since the vehicle could be servicing other customers instead of waiting for some customer to open. This suggests the design of a savings approach which accounts for both the spacial and temporal closeness of customers. The enhancement we propose in this direction is to limit the waiting time when joining two customers, i.e., if

$$w_j > W,$$

where W is a prespecified, problem dependent constant, then the link (i,j) should not be used.

3.2.2 A Time Oriented Nearest Neighbor Heuristic

This is a sequential, tour building algorithm. Starting from the depot, this heuristic finds the customer "closest" (in terms of a measure to be described later) to the last customer added to the route. This search is performed among all the customers who can feasibly (with respect to time windows and capacity constraints) be added to the end of the emerging route. A new route is started anytime the search fails, unless there are no more customers to schedule.

The metric we are using tries to account for both geographical and temporal closeness of customers. It is a measure of the distance between two customers, the time difference between their respective delivery times and the urgency of delivery to a customer, given the current point in the schedule.

Formally, if

d_{ij} = distance between customer i and customer j,

$T_{ij} = \max\{e_j, (t_i + s_i + t_{ij})\} - (t_i + s_i)$

= time interval between the departure from i and delivery at j,

$u_{ij} = d_j - (t_i + s_i + t_{ij})$

= time left until j becomes infeasible given that we are currently at i, then

$$c_{ij} = a_1 d_{ij} + a_2 T_{ij} + a_3 u_{ij}\ , \quad \text{where } a_1 + a_2 + a_3 = 1.$$

The weights attached to the above criteria are dependent on the tightness of the time windows data.

The following example underlines the importance of using our metric, rather than only a distance oriented metric. The example is illustrated in diagram 3.5. The arc weights depicted below are distances. Assume travel times equal distances.

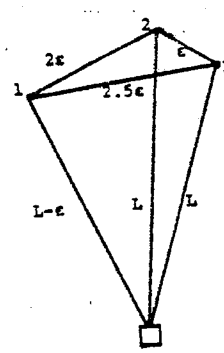


Diagram 3.5.

Let customers 1, 2, 3 have time windows [L-e, L+e], [L+2e, L+4e], [L, L+2e], respectively, where L is a large number, L>>e. We have:

$$c_{12} = a_1(2e) + a_2[(L+2e)-(L-e)] + a_3[(L+4e)-(L-e+2e)]$$
$$= (2a_1 + 3a_2 + 3a_3)e.$$
$$c_{13} = a_1(2.5e) + a_2[(L+1.5e)-(L-e)] + a_3[(L+2e)-(L-e+2.5e)]$$
$$= (2.5a_1 + 2.5a_2 + .5a_3)e.$$

If $a_1 = 1$, $a_2 = a_3 = 0$, then $c_{12} = 2e$, $c_{13} = 2.5e$, and, hence, the next customer visited will be customer 2. Since it is time infeasible to add customer 3 at the end of the route (0,1,2) ($t_2=L+2e$, $t_3=L+3e>d_3$), the vehicle must return to the depot and customer 3 will have to be serviced on a direct route. On the other hand, by letting $a_1=a_2=.4$, $a_3=.2$, we get: $c_{12}=2.6e$, $c_{13}=2.1e$, and hence our heuristic will find the optimal route (0,1,3,2,0).

3.2.3 Insertion heuristics

This class of sequential, tour building heuristics use two criteria $c_1(i,u,j)$ and $c_2(i,u,j)$ to insert a new customer into a route under construction, where i,j are adjacent customers on the route, and u is yet unrouted. Let $(i_1,i_2,\ldots,i_m)$ be the current route, where $i_1 = i_m = 0$. For each unrouted customer, we first compute its best feasible insertion place in the emerging route as:

$$c_1(i_u,u,j_u) = \underset{p=1,\ldots,m-1}{\text{optimum}} [c_1(i_p,u,i_{p+1})]$$

Inserting u between i_u and i_{u+1} could potentially alter all the arrival times at customers $(i_{u+1},\ldots,i_m)$. One can easily see that this insertion is time feasible if and only if lemma 3.2 holds for $f_1 = u$ and $S(f_1)=S(u)=\{i_{u+1},\ldots,i_m\}$. Next, the best customer u to be inserted in the route is selected as the one for which

$$c_2(i_u*,u^*,j_u*) = \underset{u \text{ unrouted and feasible}}{\text{optimum}} [c_2(i_u,u,j_u)]$$

Customer u^* is then inserted in the route between i_u* and j_u*. When no more customers with feasible insertions can be found, a new route is started, unless all customers have been routed.

Route initialization can have a significant impact on the behavior of a heuristic. Initially, every route is of the form (0,i,0). We have experimented with several criteria for selecting the first customer on the route. These are: the furthest unrouted customer, the unrouted customer with the earliest delivery deadline and the unrouted customer with the minimum weighted combination of direct route time and distance.

Based on the general insertion criteria described above, we have considered three more specific approaches.

i) $c_{11}(i,u,j) = d_{iu} + d_{uj} - a_1 d_{ij};$

$c_{12}(i,u,j) = t_{j_u} - t_j,$

where t_{j_u} is the new arrival time at j given that u is on the route;

$$c_1(i,u,j) = a_{11}c_{11}(i,u,j) + a_{12}c_{12}(i,u,j),$$

$a_{11}+a_{12}=1.$

$$c_2(i,u,j) = a_2 c_{0u} - c_1(i,u,j)$$

This type of insertion heuristics tries to maximize the benefit derived from servicing a customer on the partial route being constructed rather than on a direct route. The best feasible insertion place for an unrouted customer is the one that minimizes the weighted combination of distance and time insertion. Clearly, different values of a_1 and a_2 lead to possible different criteria for selecting the customer for insertion and its best insertion spot. For example, if $a_{12}=0$ and $c_{0u}=d_{0u}$, one obtains the Mole and Jameson(1976) approach introduced for the (VRP). Furthermore, setting $a_1=1$, $a_2=1$ creates a generalized savings criterion while $1 \leq a_2 \leq 2$, $a_1=a_2-1$, gives a generalized Gaskell's criterion (Gaskell, 1967).

Therefore, the class of heuristics described here, where insertion costs are considered on both distance and time dimensions, are a generalization of the above approaches.

A final comment should be made with respect to the cost of direct routes, c_{0u}. In general, it can be considered as the weighted combination of direct route distance and time cost. Emphasis on the distance component will encourage the early inclusion in the current partial route of customers distant from the depot, while attention to the time component could lead to the insertion of customers close to the depot but with late earliest delivery times.

The second type of insertion heuristics are directed toward the selection of customers whose insertion costs minimize a measure of total route distance and time.

ii) $c_1(i,u,j)$ is defined as before,

$$c_2(i,u,j) = a_d R_d + a_t R_t, \quad a_d + a_t = 1,$$

where R_d and R_t are current estimates of the total route distance and time of the completed route. The distance effect of an insertion on total route distance is independent of what other insertions will be performed