Proc. Natl. Sci. Counc. ROC(A)
Vol. 23, No. 4, 1999. pp. 526-536

A Method for Vehicle Routing Problem with Multiple Vehicle Types and Time Windows

FUH-HWA LIU AND SHENG-YUAN SHEN

Department of Industrial Engineering and Management
National Chiao Tung University
Hsinchu, Taiwan, R.O.C.

(Received September 3, 1998; Accepted November 27, 1998)

ABSTRACT

This paper describes a route construction method for the vehicle routing problem with multiple vehicle types and time window constraints. Several insertion-based savings heuristics are presented. The heuristics were tested on twenty-four 100-customer problems modified from the literature. Experimental results demonstrate that heuristics with the consideration of a sequential route construction parameter yielded significantly better solution quality than did all other heuristics tested.

Key Words: heterogeneous fleet, heuristics, time windows, vehicle routing

I. Introduction

The classical vehicle routing problem with multiple vehicle types (VRPMVT) is a problem of simultaneously determining the composition and routing of a heterogeneous fleet of vehicles in order to service a pre-specified set of customers with known delivery demands from a central depot. Since the temporal aspect of vehicle routing problems has become increasingly important in realistic applications, this paper extends the classical VRPMVT by imposing time window constraints on the customers and the central depot. We may also regard the VRPMVT with time windows (VRPMVTTW) as a generalization of the classical vehicle routing problem with time windows (VRPTW).

The VRPMVTTW can be formally defined as follows. Let $G=(V, A)$ be a graph with node set $V=N\cup\{0\}$ and arc set $A=\{(i,j)|i\in V, j\in V, i\neq j\}$, where $N=\{1, 2, \ldots, n\}$ denotes the customer set, and node 0 denotes the central depot. Associated with each node $i\in V$ is a demand q_i, a service time s_i and a time window (a_i,b_i) except that q_0 and s_0 are zero. A distance matrix $[\boldsymbol{d}_{i,j}]$ and a travel time matrix $[\boldsymbol{t}_{i,j}]$ are known. We assume that K different types of vehicles are available with infinite supply. For each vehicle type k, its acquisition cost and capacity are given by F_k and Q_k, respectively. Without loss of generality, each customer demand is assumed to be less than the capacity for the largest type of vehicle. Moreover, no split delivery is allowed. The objective of the VRPMVTTW is to minimize the sum of the vehicle acquisition costs and routing costs such that the following constraints are satisfied:

(1) each route begins and ends at the central depot;
(2) each customer in N is visited exactly once;
(3) the total demand of all customers served on a route can not exceed the capacity of the vehicle assigned to that route.

The time window constraints considered in this paper constitute "hard" constraints, which require that a vehicle can not visit a customer beyond a specified latest starting service time. However, a vehicle can wait if it arrives too early at a customer location.

To our knowledge, works related to the VRPMVTTW are few. Case-oriented studies can be found in Semet and Taillard (1993), Rochat and Semet (1994), and Brandão and Mercer (1997). In addition, Ferland and Michelon (1988) discussed several methods for the vehicle scheduling problem with multiple vehicle types. In contrast, both the VRPMVT and the VRPTW have received more attention during the past fifteen years. Based on the literature, heuristic methods for VRPMVT can be summarized as follows: (1) adaptations of the Clarke and Wright savings algorithm, e.g., Golden *et al.* (1984); (2) the giant tour partitioning approach, e.g., Golden *et al.* (1984); (3) the matching based savings heuristics, e.g., Desrochers and Verhoog (1991); (4) the generalized assignment based heuristic, e.g., Gheysens *et al.* (1986); (5) the sophisticated improvement based heuristic, e.g., Salhi and Rand (1993).

As to the VRPTW, Solomon (1987) found that heuristics based on adaptations of the savings algorithm and on the giant tour partitioning approach had poor performance while sequential insertion-type heuristics yielded much better results. Recent works such as those of Potvin and Rousseau (1993), Kontoravdis and Bard (1995), Russell (1995) and Liu and Shen (1998) have further indicated that parallel insertion-type heuristics are, in general, superior to sequential insertion-type heuristics. Broadly speaking, a seed generation procedure is required in order to construct routes in parallel. Customers are then inserted into one of the routes according to a specified cost metric. A local improvement procedure can be invoked after obtaining a solution or within the route construction phase. Judging from the results reported by Thompson and Psaraftis (1993), Garcia *et al*. (1994), and Potvin and Rousseau (1995), algorithms that only concentrate on improving a poor initial solution did not behave very well within a limited computation time. Though Rochat and Taillard (1995) and Taillard *et al*. (1997) obtained several best known results for Solomon's data sets using the tabu search method, the corresponding computational effort was quite high. In addition, Savelsbergh (1985) showed that finding a feasible solution to the traveling salesman problem with time windows is NP-hard. Accordingly, the generalized assignment based approach might encounter trouble in generating assignments that possess feasible solutions.

Though past results have shown the effectiveness of using insertion-type heuristics in solving the VRPTW, the sequential construction or parallel construction methods mentioned above might not be very successful in solving the VRPMVTTW. On one hand, no algorithms had solved the VRPMVT in a sequential construction way. This may reveal that the sequential route construction method is probably not an efficient way of handling vehicle routing problems with many different types of vehicles. On the other hand, estimating a good fleet composition for the VRPMVTTW is itself a difficult task when we try to construct routes in a parallel manner. Based on consideration of the possible strengths and weaknesses of the methods for the VRPMVT and the VRPTW, this paper presents several insertion-based savings heuristics for solving the VRPMVTTW. Our main purpose is to find a good initial solution to the VRPMVTTW. There is no question that a local improvement procedure can be designed to further enhance the solution quality.

The next section reviews several savings heuristics for the VRPMVT, which are closely related to our heuristics. Section III describes our methodology for solving the VRPMVTTW. Computation results are reported in Section IV, and concluding remarks are offered in Section V.

II. A Review of Savings Heuristic for the VRPMVT

The famous Clarke-Wright (CW) savings algorithm (Clarke and Wright, 1964) was originally developed to solve the classical VRP. Let $c_{i,j}$ be the traveling cost from customer i to j. For example, $c_{i,j}$ can be equal to $d_{i,j}$ or $t_{i,j}$. In the CW algorithm, each customer is initially serviced by a single vehicle. Two distinct routes containing customers i and j can be combined if i and j are the first or the last customers of their respective routes, and if the total demand of the combined route does not exceed the vehicle capacity. Clarke and Wright (1964) defined the savings that can be achieved by combining two routes into one as $S_{i,j}=c_{i,0}+c_{0,j}-c_{i,j}$. For each iteration, two routes that yield the largest savings are merged. This procedure continues until no further feasible combinations exist.

It can be seen that the CW algorithm ignores the vehicle acquisition costs. A deficient solution is likely to be obtained if the CW algorithm is directly applied to the VRPMVT. Consequently, Golden *et al*. (1984) modified the CW savings formula by considering the vehicle acquisition costs in an explicit way. We briefly state four modified savings formulas proposed by them in the following.

(1) Combined Savings (CS):
Let $F(z)$ denote the acquisition cost of the smallest vehicle that can service a route with a total demand of z. Moreover, if k is the first or the last customer of a route, we use z_k to denote the total demand currently served on this route. Clearly, two routes with total demand, respectively, of z_i and z_j can be combined by using a vehicle that costs $F(z_i+z_j)$. Thus, the combined savings due to combining the two routes is

$$S^1_{i,j}=S_{i,j}+F(z_i)+F(z_j)-F(z_i+z_j).$$

(2) Optimistic Opportunity Savings (OOS)
Because possible savings are ignored in future combinations, the CS approach tends to myopically combine two routes. The OOS approach considers the potential extra savings due to the unused capacity after two routes are combined. Let $P(z)$ be the capacity of the smallest vehicle that can service a route with a total demand of z. Golden *et al*. (1984) defined the OOS formula as follows:

$$S^2_{i,j}=S^1_{i,j}+F(P(z_i+z_j)-z_i-z_j).$$

(3) Realistic Opportunity Savings (ROS)
Even for a very small quantity of z, the OOS

formula optimistically assumes that the unused capacity z can always save one additional route that has a total demand less than or equal to the vehicle capacity $P(z)$. To avoid over-combining routes, the ROS approach revises the OOS formula in the following way. First, the opportunity savings will be included in the savings formula if a combination of two routes results in using a vehicle larger than the vehicles currently being assigned to the two routes. Second, the value of opportunity savings is the acquisition cost of the largest vehicle that has a capacity less than or equal to the unused capacity after two routes are merged. Let $F'(z)$ be the acquisition cost of the largest vehicle whose capacity is less than or equal to z. Then, the savings formula for the ROS approach can be expressed as follows:

$$S^3_{i,j}=S^1_{i,j}+\delta(\tau)F'(P(z_i+z_j)-z_i-z_j),$$

where

$$\tau=P(z_i+z_j)-P(\max(z_i,\ z_j)),$$

$$\delta(\tau)=\begin{cases}0 \text{ if } \tau=0\\ 1 \text{ if } \tau>0\,.\end{cases}$$

(4) ROS with a route shape parameter λ (ROS$_{-\lambda}$): The ROS$_{-\lambda}$ algorithm simply changes the CW savings formula to $c_{0,i}+c_{j,0}-\lambda c_{i,j}$. Thus, the savings formula for ROS$_{-\lambda}$ is given by

$$S^4_{i,j}=S^3_{i,j}+(1-\lambda)c_{i,j}.$$

In contrast to the previous three savings formulas, the ROS$_{-\lambda}$ algorithm solves each problem several times by setting λ different values. The best result of these runs is then selected as a final solution.

The above four savings algorithms focus on modifications of the CW savings formula, and they terminate with no positive savings. In all other respects, they are essentially similar to the CW savings algorithm. Notice that once two routes are combined, the savings for each possible combination may need to be recalculated. Surely, a vast number of computations can be avoided by using extra computer memory.

III. Methodology for the VRPMVTTW

It is obvious that a route may contain either only one customer or more than one customer. Let TYPE-I and TYPE-II represent the sets of routes containing only one customer and containing at least two customers, respectively. For the savings algorithms stated previously, any two routes to be combined fall into one of the following three cases: (1) TYPE-I, TYPE-I; (2) TYPE-I, TYPE-II; (3) TYPE-II, TYPE-II. An example that shows the possible combinations for each case is illustrated in Fig. 1(a)-(c). However, we can interpret these cases by using the insertion viewpoint instead of the traditional combining concept. We know that routes to be constructed for the VRPMVTTW are directed. To extend the following explanation to the VRPMVTTW, we artificially impose a direction on each route in Fig. 1(a)-(c), as shown in Fig. 1(d)-(f). For an arbitrary directed TYPE-II route, say (0-f-...-g-0), we will call (f-...-g) ***a generalized customer*** and (g-...-f) ***a reversed generalized customer***.

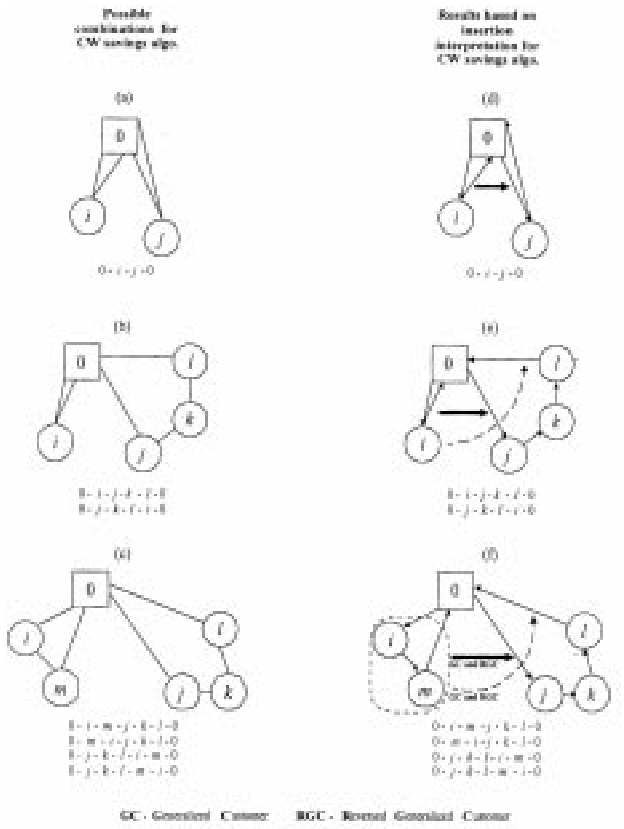

Fig. 1. Interpretations for the Clarke-Wright savings algorithm.

(1) In Fig. 1(d), it can be seen that we try to insert a route containing customer i into link (0, j) or into link (j, 0). This gives the possible resulting routes (0-i-j-0) and (0-j-i-0). In fact, these two routes are identical if we ignore the artificial direction.

(2) In Fig. 1(e), each link of (0, j) and (l, 0) can be

viewed as a candidate position into which we try to insert a route containing customer *i*. Accordingly, the possible resulting routes are (0-*i*-*j*-*k*-*l*-0) and (0-*j*-*k*-*l*-*i*-0).

(3) In Fig. 1(f), we can see that *the generalized customer* (*i*-*m*) and *the reversed generalized customer* (*m*-*i*) are each considered for insertion into the candidate links (0, *j*) and (*l*, 0). Accordingly, the possible resulting routes are (0-*i*-*m*-*j*-*k*-*l*-0), (0-*m*-*i*-*j*-*k*-*l*-0), (0-*j*-*k*-*l*-*i*-*m*-0) and (0-*j*-*k*-*l*-*m*-*i*-0).

As can be seen in Fig. 1, their possible resulting routes are exactly the same. Thus, it is obvious that we can interpret the combining operation of CW-based algorithms by employing the insertion point of view. Recall that insertion-type algorithms yield better solution quality when used to solve the VRPTW. Traditionally, all the links of each route are possible insertion positions for a standard customer. Therefore, a good way to solve the VRPMVTTW is, for a given route, to compute its possible resulting savings with respect to each link of the other routes. For instance, links (*j*,*k*) and (*k*,*l*) in Fig. 1(f) are also the possible insertion positions for the generalized customer (*i*-*m*) and the reversed generalized customer (*m*-*i*). To deal with the VRPMVTTW in this way, we first describe the feasibility conditions and our modified savings formulas in the next two sections.

1. Feasibility Conditions

To simplify the analysis, we assume that the travel time matrix $[t_{i,j}]$ satisfies the triangle inequality. Minor modifications can be used to extend the following discussion to the case in which the travel time matrix does not satisfy the triangle inequality. Furthermore, a problem with a depot time window (a_0,b_0) and a service time s_i for each $i \in N$ can be easily transformed into an equivalent problem without a depot time window and zero service time for all customers. (A transformation method that still ensures the property of the triangle inequality is given in Appendix 1.) Without loss of generality, we assume that there is no time window for the deopt, and there is no service time for each customer in the following.

Note that vehicle departure times from the central depot are decision variables. The *total schedule time* for a route refers to the difference between the vehicle arrival time at the central depot and the vehicle departure time at the central depot for a vehicle assigned to that route. The *total traveling time* for a route refers to the time of actually travel on that route. Thus, the *total waiting time* for a route is the difference between the total schedule time and the total travel time for that route. We will assume that initially the first customer on each constructed route is serviced at the earliest possible time. After the complete vehicle schedules have been created, we can compute the actual departure time for each vehicle from the central depot by eliminating any unnecessary waiting time.

Regarding the capacity feasibility condition for a pair of routes under consideration, we only require that the sum of their total demands must not exceed the largest vehicle capacity. However, checking time feasibility requires more effort. Let D_i be the vehicle departure time at customer *i* of a feasible route, and let LT_i be the latest departure time when customer *i* can be feasibly served on that route. Moreover, let *s*(*i*) refer to the immediate successor of customer *i*. The following recursive expression can be used to determine LT_i:

$$LT_i=\min\{b_i,\ LT_{s(i)}-t_{i,s(i)}\}. \qquad \textbf{(1)}$$

For the cases (TYPE-I, TYPE-I) and (TYPE-I, TYPE-II), the insertion-based interpretation simply says that we are trying to insert a standard customer into a link of the other route under consideration. It is not hard to see that we need to check the following two conditions if a route containing only customer *k* is to be inserted into a candidate link (*i*,*j*):

$$D_i+t_{i,k}\le b_k, \qquad \textbf{(2)}$$

$$\max\{a_k,\ D_i+t_{i,k}\}+t_{k,j}\le LT_j. \qquad \textbf{(3)}$$

Condition **(2)** states that the vehicle must arrive at *k* before its specified latest starting service time b_k. Condition **(3)** states that the new arrival time at *j* can not exceed its latest departure time LT_j in order to maintain the time feasibility for the successors of *j*. Although insertion of customer *k* into link (*i*,*j*) may result in a shift of arrival times for all successors of *j*, expression **(1)** guarantees that we need to check the time feasibility only for customer *j*, who will be an immediate successor of *k*. We note that if *i* is the central depot, condition **(2)** is satisfied trivially. Similarly, condition **(3)** is also satisfied if *j* is the central depot.

For the case (TYPE-II, TYPE-II), more complex operations for checking the time feasibility are needed because of the presence of *a generalized customer* and of *a reversed generalized customer*. We will first describe the operations for the generalized customer. Before going to the details, we define several notations below:

TWT_r= the total waiting time for route *r*.

D_i^{new}= the new departure time for customer i if an insertion operation is performed.

SHIFT_I= the difference between the new departure time and old departure time for customer i, i.e., $D_i^{new}-D_i$. Note that if $[t_{i,j}]$ violates the triangle inequality, the value of SHIFT_i can be negative.

Consider a TYPE-II route r^1 to be inserted into link (i,j) of a TYPE-II route r^2. Let f and g represent, respectively, the first customer and the last customer currently served on route r^1. Clearly, we have the following quantities: $D_f^{new}=\max\{a_f,\ D_i+t_{i,f}\}$ and $\text{SHIFT}_f=D_f^{new}-D_f$. With these quantities, the new departure time for customer g can be computed by using the following expression:

$$D_g^{new}=D_g+\max\{\text{SHIFT}_f-\text{TWT}_{r^1},\ 0\}. \quad \textbf{(4)}$$

Expression **(4)** states that the possible forward shift in time due to customer f may be completely or partially absorbed by the total waiting time TWT_{r^1} when D_g^{new} is computed. We now state the time feasibility conditions for the case (TYPE-II, TYPE-II) in Theorem 1 (proof omitted).

Theorem 1. Given two feasible routes r^1 and r^2, the insertion of generalized customer (f-...-g) of route r^1 into link (i,j) of route r^2 is feasible if the following conditions are satisfied:

(1) $D_f^{new}\le LT_f$,

(2) $D_g^{new}+t_{g,j}\le LT_j$.

Therefore, the feasibility with respect to the time window constraints can be checked in constant time. When the reversed generalized customer of an arbitrary route is considered, we need first to see whether the corresponding route examined in the reverse direction is still time feasible. The other operations are similar to the insertion of a generalized customer.

2. Modified Savings Formulas

The savings formulas for the VRPMVT described previously consider both the spatial costs and the vehicle acquisition costs. However, temporal constraints should not be ignored in solving the VRPMVTTW because the route feasibility may be strongly affected by the time windows. To be less myopic, the temporal restriction should be a factor in determining which two routes will be selected. We next describe the modified savings formulas for each of the three different cases in the following.

Case (1). Insertion of a TYPE-I route into link $(i^{\text{II}},j^{\text{II}})$ of a TYPE-II route.

Let k^{I} be the customer of a given TYPE-I route. For the cost of actual travel, this operation will reduce the cost by $2c_{0,k^{\text{I}}}+c_{i^{\text{II}},j^{\text{II}}}$ while paying an extra cost $c_{i^{\text{II}},k^{\text{I}}}+c_{k^{\text{I}},j^{\text{II}}}$. Thus, the net savings is $2c_{0,k^{\text{I}}}+c_{i^{\text{II}},j^{\text{II}}}-c_{i^{\text{II}},k^{\text{I}}}-c_{k^{\text{I}},j^{\text{II}}}$. One possible way to maintain route feasibility for future insertions is for customer j^{II}, to minimize its pushed forward time due to the insertion of customer k^{I}. Therefore, we can compute the temporal opportunity cost by using $D_{j^{II}}^{new}-D_{j^{\text{II}}}$. In contrast to the CW savings $S_{i,j}$, our modified CW-savings (MS) for the VRPMVTTW is given by

$$\text{MS}(i^{\text{II}},k^{\text{I}},j^{\text{II}})=w(2c_{0,k^{\text{I}}}+c_{i^{\text{II}},j^{\text{II}}}-c_{i^{\text{II}},k^{\text{I}}}-c_{k^{\text{I}},j^{\text{II}}})$$
$$-(1-w)(D_{j^{II}}^{new}-D_{j^{\text{II}}}), \quad \textbf{(5)}$$

where

$$0\le w\le 1.$$

To consider the possible savings in vehicle acquisition costs, we define other modified formulas in a manner similar to the methods used in savings formulas for the VRPMVT as follows:

Modified Combined Savings (MCS)

$$\text{MCS}(i^{\text{II}},k^{\text{I}},j^{\text{II}})=\text{MS}(i^{\text{II}},k^{\text{I}},j^{\text{II}})+F(z^{\text{I}})+F(z^{\text{II}})-F(z^{\text{I}}+z^{\text{II}}); \quad \textbf{(6)}$$

Modified Optimistic Opportunity Savings (MOOS)

$$\text{MOOS}(i^{\text{II}},k^{\text{I}},j^{\text{II}})=\text{MCS}(i^{\text{II}},k^{\text{I}},j^{\text{II}})+F(P(z^{\text{I}}+z^{\text{II}})-z^{\text{I}}-z^{\text{II}}); \quad \textbf{(7)}$$

Modified Realistic Opportunity Savings (MROS)

$$\text{MROS}(i^{\text{II}},k^{\text{I}},j^{\text{II}})=\text{MCS}(i^{\text{II}},k^{\text{I}},j^{\text{II}})$$
$$+\delta(\tau)F'(P(z^{\text{I}}+z^{\text{II}})-z^{\text{I}}-z^{\text{II}}); \quad \textbf{(8)}$$

MROS with a route shape parameter λ

$$\text{MROS}_{-\lambda}(i^{\text{II}},k^{\text{I}},j^{\text{II}})=\text{MROS}(i^{\text{II}},k^{\text{I}},j^{\text{II}})+w(\lambda-1)c_{i^{\text{II}},j^{\text{II}}}. \quad \textbf{(9)}$$

Case (2). Insertion of a TYPE-I route into a link of another TYPE-I route.

This situation is a special instance of **Case (1)** if we let either i or j in Eqs. (5)-(9) be the central depot. For the purpose of later use, we denote the related savings formulas for this case by replacing the superscripts I and II in Eqs. (5)-(9) with the respective notations I_1 and I_2. For example, the modified combined savings will be given by

$$\mathrm{MCS}(i^{\mathrm{I}_2},k^{\mathrm{I}_1},j^{\mathrm{I}_2})=\mathrm{MS}(i^{\mathrm{I}_2},k^{\mathrm{I}_1},j^{\mathrm{I}_2})+F(z^{\mathrm{I}_1})+F(z^{\mathrm{I}_2})$$

$$-F(z^{\mathrm{I}_1}+z^{\mathrm{I}_2}).$$

Case (3). Insertion of generalized customer (f^{II_1}-...-g^{II_1}) of a TYPE-II route into link (i^{II_2},j^{II_2}) of a distinct TYPE-II route.

For the cost of actual travel, this operation will save an amount of $c_{0,f^{\mathrm{II}_1}}+c_{g^{\mathrm{II}_1},0}+c_{i^{\mathrm{II}_2},j^{\mathrm{II}_2}}$ by paying an extra cost $c_{i^{\mathrm{II}_2},f^{\mathrm{II}_1}}+c_{g^{\mathrm{II}_1},j^{\mathrm{II}_2}}$. Thus, the net savings is

$$c_{0,f^{\mathrm{II}_1}}+c_{g^{\mathrm{II}_1},0}+c_{i^{\mathrm{II}_2},j^{\mathrm{II}_2}}-c_{i^{\mathrm{II}_2},f^{\mathrm{II}_1}}-c_{g^{\mathrm{II}_1},j^{\mathrm{II}_2}}.$$

In contrast to the insertion of a standard customer, the insertion of a generalized customer tends to result in a bigger pushed forward time for customer j^{II_2}. To be on an equal computation basis, we change the temporal opportunity cost used in the above two cases to $(D^{new}_{j^{\mathrm{II}_2}}-D_{j^{\mathrm{II}_2}})/cus^{\mathrm{II}_1}$, where cus^{II_1} is the number of standard customers that composes the generalized customer (f^{II_1}-...-g^{II_1}). Accordingly, the modified CW savings for the case (TYPE-II, TYPE-II) can be expressed as follows:

$$\mathrm{MS}(i^{\mathrm{II}_2},f^{\mathrm{II}_1},g^{\mathrm{II}_1},j^{\mathrm{II}_2})$$

$$=w(c_{0,f^{\mathrm{II}_1}}+c_{g^{\mathrm{II}_1},0}+c_{i^{\mathrm{II}_2},j^{\mathrm{II}_2}}-c_{i^{\mathrm{II}_2},f^{\mathrm{II}_1}}-c_{g^{\mathrm{II}_1},j^{\mathrm{II}_2}})-(1-w)$$

$$\times(D^{new}_{j^{\mathrm{II}_2}}-D_{j^{\mathrm{II}_2}})/cus^{\mathrm{II}_1}.$$

All other modified savings formulas for this case can be obtained in a way similar to that used in **Case (1)**. For example, the modified combined savings is given by

$$\mathrm{MCS}(i^{\mathrm{II}_2},f^{\mathrm{II}_1},g^{\mathrm{II}_1},j^{\mathrm{II}_2})$$

$$=\mathrm{MS}(i^{\mathrm{II}_2},f^{\mathrm{II}_1},g^{\mathrm{II}_1},j^{\mathrm{II}_2})+F(z^{\mathrm{II}_1})+F(z^{\mathrm{II}_2})-F(z^{\mathrm{II}_1}+z^{\mathrm{II}_2}).$$

3. Selection Criteria for Insertion

With a total of n partial routes initially constructed by using the smallest type of vehicle feasible to service one customer only, we will now try to determine which route will be inserted into a link of another route. Our heuristics repeat the following selection and insertion steps until no feasible combinations can yield positive savings. For convenience, the modified combined savings formula will be used below for the purpose of explanation.

Step 1. For each case, (TYPE-I, TYPE-I), (TYPE-I, TYPE-II), and (TYPE-II, TYPE-II), compute their respective maximal savings, i.e.,

$$\mathrm{MCS}(i^{*\mathrm{I}_2},k^{*\mathrm{I}_1},j^{*\mathrm{I}_2})=\max\{\mathrm{MCS}(i^{\mathrm{I}_2},k^{\mathrm{I}_1},j^{\mathrm{I}_2})\};$$

$$\mathrm{MCS}(i^{*\mathrm{II}},k^{*\mathrm{I}},j^{*\mathrm{II}})=\max\{\mathrm{MCS}(i^{\mathrm{II}},k^{\mathrm{I}},j^{\mathrm{II}})\};$$

$$\mathrm{MCS}(i^{*\mathrm{II}_2},f^{*\mathrm{II}_1},g^{*\mathrm{II}_1},j^{*\mathrm{II}_2})=\max\{\mathrm{MCS}(i^{\mathrm{II}_2},f^{\mathrm{II}_1},g^{\mathrm{II}_1},j^{\mathrm{II}_2})\}.$$

If a given route can not be feasibly inserted into a link of the other route under consideration, a negative infinity number corresponding to this insertion operation is given. Note that all candidate links will be considered.

Step 2. At each iteration, the best insertion operation to be carried out is the one with the largest savings determined by the following expression:

$$\max\{\mathrm{MCS}(i^{*\mathrm{I}_2},k^{*\mathrm{I}_1},j^{*\mathrm{I}_2}),\ \eta\times\mathrm{MCS}(i^{*\mathrm{II}},k^{*\mathrm{I}},j^{*\mathrm{II}}),$$

$$\mathrm{MCS}(i^{*\mathrm{II}_2},f^{*\mathrm{II}_1},g^{*\mathrm{II}_1},j^{*\mathrm{II}_2})\},$$

where the parameter η represents the degree of sequential construction of TYPE-II routes. The reason for using the parameter η is that constructing too many TYPE-II routes in an earlier stage may yield poor solution quality due to the time window constraints. When we are trying to use larger vehicles (or fewer vehicles) to service the pairs of short TYPE-II routes, time window constraints might cause infeasibility even though the cost would be reduced by a large amount. How the values of η will be set will be evaluated through our computational studies. If we set η to a very large number, this may show that we prefer to insert a TYPE-I route into a link of a TYPE-II route if feasibility is maintained. In other words, we do not attempt to create another TYPE-II route until the existing TYPE-II route is not feasible for all TYPE-I routes. In this case, we would construct TYPE-II routes in a sequential manner, similar to the traditional sequential route construction method for solving the VRPTW.

Thus, it is clear that we have used three design parameters, w, λ and η, in developing our insertion-based savings heuristics. To obtain higher solution quality, we may run the heuristics several times by setting different values for these parameters. For a single run, the time complexity of our heuristics can be analyzed as follows. As we have stated before, a total of n initial routes are constructed first. We then try to determine which route is to be inserted into a link of another route. Clearly, each route has $O(n)$ ways in which a link can be selected for insertion. Because checking the feasibility with respect to the time window constraints can be done in constant time, the complexity of computing the savings for all links is $O(n^2)$. Therefore, the complexity of finding the best insertion place among all the links is also $O(n^2)$. After the insertion of a customer or a (reversed) generalized customer into link (i,j) of a route, we need to update the latest departure time LT for each predecessor of j; that is, a time on the order of n is required. Accordingly, the complexity of one iteration for performing an insertion operation of our heuristic is $O(n^2)$. Since there are at most n–1 iterations to be performed, the overall complexity of a single run of our heuristic is $O(n^3)$.

IV. Computational Results

To evaluate the performance of our heuristics, we have modified the savings algorithms for the VRPMVT. Because routes constructed for the VRPMVTTW are directed, the CW savings, $S_{i,j}=c_{i,0}+c_{0,j}-c_{i,j}$, will imply that we can save cost $S_{i,j}$ through the combination of a route containing the *last-serviced* customer i and a route containing the *first-serviced* customer j. We only replace the corresponding basic component $S_{i,j}$ in the savings formulas for VRPMVT with the following expression:

$$S'_{i,j}=w(c_{i,0}+c_{0,j}-c_{i,j})-(1-w)(D_j^{new}-D_j)/cus^{r(i)},$$

where $cus^{r(i)}$ is the number of customers for a route containing i. In addition, two other savings formulas, $CS_{-\lambda}$ and $OOS_{-\lambda}$ similar to $ROS_{-\lambda}$, were also studied in our experiment.

The data associated with 24 sample problems consisting of 100 customers are given in Appendix 2. The first twelve problems have a shorter scheduling horizon, and the last twelve problems have a longer scheduling horizon. Customer locations for problems 1-4 and 13-16 are randomly distributed. Problems 5-8 and 17-20 are clustered customers. The remaining problems have a mix of randomly dispersed and clustered customers. In our test, the weight parameter w varied from 0.0 to 1.0 by 0.25, and the route shape parameter λ varied from 0.0 to 3.0 by 0.5. As to the parameter η, the degree of sequential construction of TYPE-II routes, four values (1.0, 1.5, 2.0, 99) were tested.

All the programs were coded in Fortran 90 and compiled using Microsoft PowerStation 4.0 on a Pentium II 233 personal computer, and the computations were performed using real arithmetic. In addition, the travel time was set to be the distance traveled. It is clear that the total amount of service time is a fixed constant for any algorithm. In order to compare the performance of different heuristics on an unbiased platform, for each problem, the total cost reported in the following tables did not include the service times. The total cost of routing refers to the summation of the total scheduled time for each route.

Tables 1 and 2 compare the performance of those heuristics originally designed for VRPMVT with our heuristics without considering the parameter η, i.e., η=1. It can be seen from Tables 1 and 2 that our heuristics, even without considering adjustment of parameter η, still yielded better solution quality on an average of 16% to 23% than did those heuristics based on the combining concept. Among 144 pairs of solutions in Tables 1 and 2, only in one case for problem 8 did our heuristics have relatively poor performance. The results may provide support for our argument that algorithms based on the combining concept can not handle problems with time windows in a very efficient way. In particular, we achieved significantly improved results for the last twelve problems. This phenomenon indicates that our heuristics behaved more effectively for problems with a long scheduling horizon. It further demonstrates that algorithms based on the combining concept tend to use more vehicles (or yield shorter routes) without taking advantage of an existing long scheduling horizon. For convenience, Table 4 summarizes the computation times associated with the results in Tables 1-3.

Table 3 reports the results for our heuristics with the settings for λ and η. The columns labeled η give the minimal setting that corresponds to the best solution of each problem for each heuristic. It can be found that most of the solutions in Table 3 are much better than the respective solutions in Tables 1 and 2. This shows that the use of parameter η substantially enhances solution quality. Each of the heuristics, $MCS_{-\lambda-\eta}$, $MOOS_{-\lambda-\eta}$, and $MROS_{-\lambda-\eta}$, produced the best solution in 12, 6, and 16 out of the 24 problems, respectively. Note that $MCS_{-\lambda-\eta}$ and $MROS_{-\lambda-\eta}$ both produced the best solutions in 11 problems. However, further experiments are needed to investigate this interesting result.

To investigate the effect of different values of η on the solution quality, in addition to η=99, we allowed η to vary from 1.0 to 3.0 in increments of 0.2. Figure

Table 1. Results for Heuristics without Parameters λ and η

Problem	(1)CS	(2)MCS	(1)/(2)	(3)OOS	(4)MOOS	(3)/(4)	(5)ROS	(6)MROS	(5)/(6)
1	6696	6135	1.09	6730	6137	1.10	6696	6135	1.10
2	6089	5648	1.08	5997	5565	1.08	6089	5648	1.06
3	5250	4933	1.06	5282	4943	1.07	5250	4933	1.07
4	4672	4575	1.02	4701	4618	1.02	4672	4536	1.04
5	11442	11361	1.01	11751	10692	1.10	11442	11361	1.03
6	9528	9393	1.01	9662	8788	1.10	9528	9393	1.03
7	9188	9079	1.01	9533	9315	1.02	9188	9079	1.05
8	7987	7754	1.03	8641	8702	0.99	7987	7754	1.11
9	6910	6413	1.08	7091	6420	1.10	6854	6340	1.12
10	6539	6154	1.06	6740	6014	1.12	6505	6084	1.11
11	5974	5774	1.03	5999	5639	1.06	5930	5596	1.07
12	5414	5356	1.01	5526	5365	1.03	5429	5318	1.04
13	16655	8064	2.07	16655	8064	2.07	16655	8064	2.07
14	12652	6967	1.82	12652	6967	1.82	12652	6967	1.82
15	9719	6803	1.43	9719	6803	1.43	9719	6803	1.43
16	5925	4929	1.20	5925	4929	1.20	5925	4929	1.20
17	11058	9815	1.13	11058	9815	1.13	11058	9815	1.13
18	15559	10765	1.45	15559	9580	1.62	15559	10765	1.45
19	13299	9316	1.43	13299	9316	1.43	13299	9316	1.43
20	8350	7240	1.15	8350	7240	1.15	8350	7240	1.15
21	12953	10076	1.29	12953	9963	1.30	12953	10076	1.29
22	10583	8995	1.18	10744	8661	1.24	10583	8995	1.19
23	8630	7961	1.08	8797	7488	1.17	8630	7961	1.11
24	5312	5054	1.05	6011	5199	1.16	5312	5054	1.19
Average			1.20			1.23			1.22

Table 2. Results for Heuristics with Parameter λ

Problem	(1)CS$_{-\lambda}$	(2)MCS$_{-\lambda}$	(1)/(2)	(3)OOS$_{-\lambda}$	(4)MOOS$_{-\lambda}$	(3)/(4)	(5)ROS$_{-\lambda}$	(6)MROS$_{-\lambda}$	(5)/(6)
1	6652	5750	1.16	6700	5826	1.15	6652	5750	1.16
2	6029	5447	1.11	5961	5429	1.10	6029	5447	1.11
3	5156	4933	1.05	5210	4943	1.05	5156	4933	1.05
4	4576	4455	1.03	4544	4528	1.00	4576	4455	1.03
5	11040	10743	1.03	11273	9247	1.22	11040	10743	1.03
6	9190	9153	1.00	9662	8788	1.10	9190	9153	1.00
7	9014	8805	1.02	9533	9031	1.06	9014	8805	1.02
8	7867	7481	1.05	8477	8449	1.00	7867	7481	1.05
9	6840	6191	1.10	7025	6150	1.14	6844	6096	1.12
10	6263	5942	1.05	6363	5870	1.08	6343	5994	1.06
11	5824	5581	1.04	5918	5491	1.08	5800	5596	1.04
12	5380	5328	1.01	5318	5303	1.00	5324	5318	1.00
13	9582	8064	1.19	9582	8064	1.19	9582	8064	1.19
14	9679	6967	1.39	9679	6967	1.39	9679	6967	1.39
15	8531	5928	1.44	8531	5928	1.44	8531	5928	1.44
16	4850	4176	1.16	5074	4176	1.22	4850	4176	1.16
17	11058	6711	1.65	11058	7354	1.50	11058	6711	1.65
18	11944	8177	1.46	11944	8177	1.46	11944	8177	1.46
19	11076	8074	1.37	11076	8074	1.37	11076	8074	1.37
20	7311	6744	1.08	7311	6744	1.08	7311	6744	1.08
21	10572	9330	1.13	10572	8474	1.25	10572	9330	1.13
22	10339	8856	1.17	10405	8013	1.30	10339	8856	1.17
23	8180	7240	1.13	8797	7120	1.24	8180	7240	1.13
24	5312	5054	1.05	5731	5199	1.10	5312	5054	1.05
Average			1.16			1.19			1.16

Table 3. Results for Heuristics with Parameters λ and η

Problem	$MCS_{-\lambda-\eta}$		$MOOS_{-\lambda-\eta}$		$MROS_{-\lambda-\eta}$		Best fleet mix
	Cost	η	Cost	η	Cost	η	
1	5090	99.0	5169	2.0	5061*	99.0	$A^1B^{11}C^{11}D^1$
2	5017	99.0	5046	2.0	5013*	99.0	$A^1B^4C^{14}D^2$
3	4772*	2.0	4852	2.0	4772*	2.0	B^7C^{15}
4	4455*	1.0	4528	1.0	4455*	1.0	B^9C^{14}
5	9272	1.5	9247*	1.0	9272	1.5	A^1B^{10}
6	8433*	2.0	8702	1.5	8433*	2.0	A^{19}
7	8033*	1.5	8703	1.5	8033*	1.5	A^{19}
8	7384*	1.5	8416	1.5	7384*	1.5	A^{19}
9	5813	99.0	5957	2.0	5687*	99.0	$A^7B^7C^7$
10	5702	99.0	5671	2.0	5649*	99.0	$A^5B^6C^8$
11	5504	99.0	5491	1.0	5419*	99.0	$A^{11}B^2C^8$
12	5189*	99.0	5303	1.0	5318	1.0	$A^2B^{13}C^3D^1$
13	4593*	1.5	4674	1.5	4593*	1.5	A^5
14	4331*	1.5	4681	1.5	4331*	1.5	A^5
15	4220*	1.5	4368	1.5	4220*	1.5	A^4B^1
16	3849*	1.5	3910	1.5	3849*	1.5	A^5
17	6711*	1.0	7354	1.0	6711*	1.0	A^4B^1
18	7720*	99.0	7720*	1.5	7720*	99.0	A^1C^3
19	7466*	99.0	7466*	1.5	7466*	99.0	C^2D^1
20	6744*	1.0	6744*	1.0	6744*	1.0	A^5
21	5969	99.0	5871*	99.0	5969	99.0	C^1E^3
22	6228	99.0	5945*	99.0	6228	99.0	$A^1C^1D^1E^2$
23	5968*	99.0	5790	99.0	5968*	99.0	$A^1B^1C^5$
24	4983*	1.5	5199	1.0	4983*	1.5	$A^{14}B^2$

Table 4. Summary of Computation Times (in Seconds)

Prob.	No λ and η						With λ						With λ and η		
	CS	MCS	OOS	MOOS	ROS	MROS	CS	MCS	OOS	MOOS	ROS	MROS	MCS	MOOS	MROS
1	1	1	1	1	1	1	5	7	8	9	10	11	26	33	39
2	2	2	2	2	2	2	7	9	11	12	13	15	35	45	54
3	1	2	2	2	2	3	8	11	12	15	15	18	41	54	66
4	1	2	2	3	2	3	9	13	14	18	16	21	48	63	76
5	1	1	1	1	1	1	6	7	8	9	9	10	27	34	38
6	1	1	1	1	2	2	7	9	10	11	11	13	34	42	49
7	1	2	1	2	2	3	9	11	11	13	12	15	40	50	57
8	1	2	2	2	1	2	9	11	12	14	13	16	43	56	63
9	1	1	1	1	1	2	6	7	9	9	10	10	26	34	39
10	1	1	1	1	1	2	7	9	10	12	11	14	33	43	50
11	2	1	2	2	2	2	8	10	11	15	12	17	39	52	61
12	1	2	2	3	2	3	9	13	12	18	14	19	46	63	71
13	1	1	2	2	1	2	7	9	9	11	10	12	33	40	45
14	1	1	1	2	2	2	8	10	11	13	12	14	38	48	53
15	1	1	2	2	2	2	9	11	12	14	13	16	42	54	60
16	1	2	2	2	1	2	9	12	12	15	13	16	45	59	66
17	1	1	2	2	2	1	6	7	8	9	9	10	28	35	39
18	1	1	1	2	2	2	8	10	11	11	11	13	36	45	50
19	2	1	2	2	2	2	9	11	11	13	13	16	41	52	59
20	1	1	2	2	2	2	9	12	12	15	13	16	45	57	64
21	1	1	1	2	2	2	7	9	9	11	10	12	33	42	47
22	1	2	2	1	2	2	8	10	10	13	12	14	39	49	56
23	1	1	2	3	2	2	9	12	12	15	13	16	44	57	63
24	1	2	2	2	2	3	9	13	13	17	14	16	51	64	72

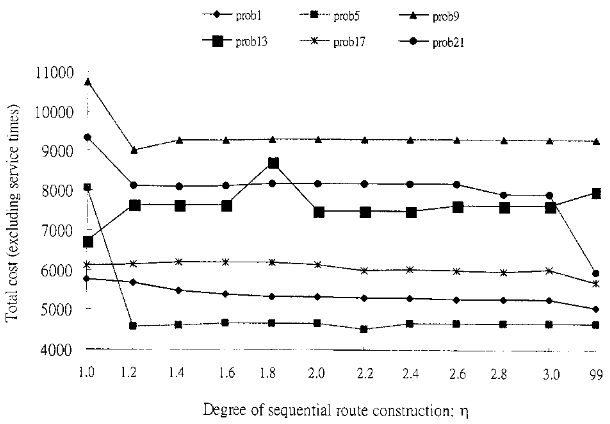

Fig. 2. The effect of η on the solution quality.

2 displays the results for applying MROS$_{-\lambda-\eta}$ to problems 1, 5, 9, 13, 17 and 21. Excluding the two extreme values of η=1 and η=99, we can see that the results that correspond to other values of η did not significantly fluctuate except for problem 13 at η=1.8. Thus, this may mean that we do not need to specify many different values for η. Though the best setting for η can vary for a specific problem, the values of (1, 1.5, 2.0, 99) for η yielded satisfactory results according to the results of our empirical test. Table 3 shows that our heuristics found the best result at η=1 in 14 out of the 72 solutions and at η=99 in 24 out of the 72 solutions. Thus, we suggest that two extreme values of η=1 and a very large number, e.g., 99 in our test, should be included in the settings for η.

V. Conclusions

Both the VRPMVT and the VRPTW are important and practical problems in real world applications. This paper has described several insertion-based savings heuristics for solving the VRPMVT with time windows (or the VRPTW with a heterogeneous fleet). The major idea underlying our method is a generalization of the traditional insertion viewpoint instead of the combining concept used in the CW savings-based heuristics. In order to avoid constructing too many short routes, a parameter η that represents the degree of sequential route construction has been considered and verified. Experimental results appear to show that our insertion-based savings strategy is encouraging. For all the heuristics tested, the heuristics in which the parameter η is considered were found to be most successful.

From a practical viewpoint, the VRPMVTTW is worth further study. A future research direction may be to develop sophisticated improvement procedures for obtaining higher solution quality. Another direction is to develop an efficient method for finding a tight lower bound for the VRPMVTTW.

Appendix 1
Transformation Method

A VRPTW instance π with depot time window (a_0, b_0), service time $s_i \neq 0$ for all $i \in N$ and a travel time matrix $[t_{i,j}]$ satisfying the triangle inequality can be transformed into an equivalent instance π' without a depot time window and zero service time for all customers. Surely, the property of triangle inequality should be maintained. A simple transformation is to define the new travel times from i to j $(i,j \in V)$ as $t'_{i,j} = t_{i,j} + s_i/2 + s_j/2$ and the new time window for customer i as $(a'_i, b'_i) = (\max\{a_i + s_i/2,\ a_0 + t_{0,i} + s_i/2\},\ \min\{b_i + s_i/2,\ b_0 - t_{i,0} - s_i/2\})$. It is obvious that the triangle inequality for the new travel time matrix $[t'_{i,j}]$ is satisfied.

Appendix 2
Problem Data

All 24 problems are modified from the Solomon (1987) benchmark sets. Corresponding problem names in the original paper are in parentheses.

#1~#4 (R101~R104)

Vehicle	Capacity	Cost
A	30	50
B	50	80
C	80	140
D	120	250
E	200	500

#5~#8 (C101~C104)

Vehicle	Capacity	Cost
A	100	300
B	200	800
C	300	1350

#9~#12 (RC101~RC104)

Vehicle	Capacity	Cost
A	40	60
B	80	150
C	150	300
D	200	450

#13~#16 (R201~R204)

Vehicle	Capacity	Cost
A	300	450
B	400	700
C	600	1200
D	1000	2500

#17~#20 (C201~C204)

Vehicle	Capacity	Cost
A	400	1000
B	500	1400
C	600	2000
D	700	2700

#21~#24 (RC201~RC204)

Vehicle	Capacity	Cost
A	100	150
B	200	350
C	300	550
D	400	800
E	500	1100
F	1000	2500

References

Brandãe, J. and A. Mercer (1997) A tabu search algorithm for the multi-trip vehicle routing and scheduling problem. *Eur. J. Opl. Res.*, **100**, 180-191.

Clarke, G. and J. Wright (1964) Scheduling of vehicles from a central depot to a number of delivery points. *Opns. Res.*, **12**, 568-581.

Desrochers, M. and T. W. Verhoog (1991) A new heuristic for the fleet size and mix vehicle routing problem. *Comps. & Opns. Res.*, **18**, 263-274.

Ferland, J. A. and P. Michelon (1988) The vehicle scheduling problem with multiple vehicle types. *J. Opl. Res. Soc*., **39**, 577-583.

Garcia, B. L., J. Y. Potvin, and J. M. Rousseau (1994) A parallel implementation of the tabu search heuristic for vehicle routing problems with time window constraints. *Comps. & Opns. Res*., **21**, 1025-1033.

Gheysens, E., B. Golden, and A. Assad (1986) A new heuristic for determining fleet size and composition. *Math. Prog. Studies*, **26**, 233-236.

Golden, B., A. Assad, L. Levy, and E. Gheysens (1984) The fleet size and mix vehicle routing problem. *Comps. & Opns. Res*., **11**, 49-66.

Kontoravdis, G. and J. F. Bard (1995) A GRASP for the vehicle routing problem with time windows. *ORSA J. Computing*, **7**, 10-23.

Liu, F. H. and S. Y. Shen (1998) A route-neighborhood-based metaheuristic for vehicle routing problem with time windows. *Eur. J. Opl. Res*. (in press).

Potvin, J. Y. and J. M. Rousseau (1993) A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. *Eur. J. Opl. Res*., **66**, 331-340.

Potvin, J. Y. and J. M. Rousseau (1995) An exchange heuristic for routing problems with time windows. *J. Opl. Res. Soc*., **46**, 1433-1446.

Rochat, Y. and F. Semet (1994) A tabu search approach for delivering pet food and flour in Switzerland. *J. Opl. Res. Soc*., **45**, 1233-1246.

Rochat, Y. and E. Taillard (1995) Probabilistic diversification and intensification in local search for vehicle routing. *J. Heuristics*, **1**, 147-167.

Russell, R. A. (1995) Hybrid heuristics for the vehicle routing problem with time windows. *Transport. Sci*., **29**, 156-166.

Salhi, S. and G. K. Rand (1993) Incorporating vehicle routing into the vehicle fleet composition problem. *Eur. J. Opl. Res*., **66**, 313-330.

Savelsbergh, M. W. P. (1985) Local search in routing problems with time windows. *Ann. Opns. Res*., **4**, 285-305.

Semet, F. and E. Taillard (1993) Solving real-life vehicle routing problems efficiently using tabu search. *Ann. Opns. Res*., **41**, 469-488.

Solomon, M. M. (1987) Algorithms for the vehicle routing and scheduling with time window constraints. *Opns. Res*., **15**, 254-265.

Taillard, E., P. Badeau, M. Gendreau, F. Guertin, and J. Y. Potvin (1997) A tabu search heuristic for the vehicle routing problem with soft time windows. *Tansport. Sci*., **31**, 170-186.

Thompson, P. M. and H. N. Psaraftis (1993) Cyclic transfer algorithms for multivehicle routing and scheduling problems. *Opns. Res*., **41**, 935-946.

時窗限制之多車種車輛途程與排程方法

劉復華　申生元

國立交通大學工業工程與管理學系

摘　要

本文就顧客具有服務時窗要求之下，物流配送業者如何以多種不同車輛，在最經濟的成本之下完成配送任務進行探討。文中提出數個具有時窗限制之多車種車輛途程與排程問題啟發式解法。爲評估所提解法，我們修正文獻中未曾考慮時窗限制的多車種車輛途程與排程問題之節省啟發式解法以進行比較。測試結果顯示我們的方法明顯優於文獻中未曾考慮時窗限制的多車種之節省啟發式解法。