Hybrid Genetic Algorithms for Vehicle Routing Problems with Time Windows

Noor Hasnah Moin

Institute of Mathematical Sciences
University of Malaya
50603 Kuala Lumpur
Malaysia.

Abstract

Blanton and Wainwright (1993) first introduced the application of Genetic Algorithm in Vehicle Routing Problem with Time Windows (VRPTW). Coupled with several problem specific crossover operators the algorithm achieved satisfactory results for the problems tested. However their methods have some drawbacks. In most instances the algorithm often do not converge to a feasible solution. To overcome this problem, in this paper hybrid genetic algorithms that incorporate heuristic methods developed by Solomon (1987) are proposed. Instead of using lexicographic ordering the problem is modelled as a multi-objective optimisation problem so that several alternative solutions can be selected from the set of final solutions. It is found that these algorithms, which incorporate a modified enhanced edge-recombination operator, are superior to those proposed by Blanton and Wainwright in all the problems tested. The hybrid genetic algorithm is also extended to solve several benchmark problems and the algorithms have produced very competitive results when compared to the best solutions found in the literature.

1. Introduction

Most real world problems encountered in distribution have a time constraint within which distribution of goods or services can be made. This is often characterised by the working pattern of the organisations/ companies, which usually operate in a fixed time schedule. In addition, customers' preferences, such as in restaurants where deliveries are only allowed before a certain time of the day, may also restrict the schedule of the vehicles involved. Normally, these issues are simplified and formulated as Vehicle Routing Problem (VRP); the solution to this relatively unconstrained problem may not be practical and is rarely implemented in real life problems without major revisions being carried out (Bodin, 1990). Although some solutions to VRP have been adapted successfully to practical routing and scheduling problems with time restrictions, the need to address these constraints explicitly in the modelling can no longer be ignored.

Vehicle Routing Problem with Time Windows (VRPTW) has only recently received attention from the research community and most of the earlier investigations in this field have concentrated on case studies developing *ad-hoc* procedures for each individual problem (Knight and Hofer (1968) and Pullen and Webb (1967)). VRPTW is an extension of VRP that not only addresses the spatial but also the temporal aspects of vehicle movement. VRPTW involves finding the best routing schedule for a fleet of homogeneous (or heterogeneous) vehicles, originating and terminating at a central depot, with limited capacities and associated maximum travel times, to service a set of customers with known demands and time windows characterised by the earliest and the latest allowable times within which the service should begin. Owing to working regulations that may be applied to drivers, a limit on the total time allowed for any vehicle may also be added and this is often attained by defining a time window at the depot. Therefore VRPTW comprises two parts, the routing and the scheduling of vehicles. The routing is concerned with finding an optimal visiting sequence of the customers whilst the scheduling specifies the time the customers are serviced. In the presence of time windows, the total cost of routing not only includes the total travel distance and the service times, but also the total cost of waiting incurred when a vehicle arrives too early at a customer location. In some instances, the objective value also incorporates the total penalty accumulated when the customers are serviced outside their time windows.

The VRPTW arises in many practical decision making problems such as: retail distribution, school bus routing, bank and postal deliveries, industrial refuse collection, newspaper delivery, fuel oil delivery, dial-a-ride service, airline and railway fleet routing and scheduling, etc. An excellent review is presented by Desrochers et al (1988), Solomon and Desrosiers (1988) and most recently by Desrosiers et al (1995).

In this paper we present two hybrid genetic algorithms developed to solve VRPTW. We will show how these algorithms can be modified appropriately to accommodate the added complexity and a comparison with two of the heuristics developed in Solomon (1987) is also made to give an indication of the quality of solutions that can be achieved using GA.

In the next section, a review of some of the solution procedures developed in the literature, with a particular emphasis on heuristic methods, is presented. The necessary notations and terminology are given in section 3.0. Before outlining our methods, we discussed two heuristic methods for VRPTW based on the work of Solomon (1987). These techniques will be embedded in our new approaches and the results obtained will be compared with those found by our new algorithms. Section 4.0 outlines our proposed methods and the results and discussion are presented in the subsequent section. Preliminary experiments carried out on 8 test problems and 6 of the benchmark problems are given and future work and suggestions regarding various modifications are discussed in the concluding remarks.

2. A Brief Review of VRPTW Heuristics

VRP is itself ℵ℘-hard and by restriction, because of the added complexity, VRPTW is also ℵ℘-hard (Solomon, 1987).

In fact, even finding a feasible solution to the VRPTW, when the number of vehicles is fixed, is itself an $\aleph\wp$-complete problem (by inference from the work of Savelsbergh (1984) on the TSPTW).

Although several exact algorithms have been proposed, it is very unlikely that solutions to realistic size problems will be found owing to their huge computational requirements (Desrochers et al (1992)).

Owing to the inherent complexity of VRPTW, the use of classical methods, such as those described above, is only successful for certain types of problems and performs poorly on others. Metaheuristics offer a great advantage over classical methods. They combine various features from different procedures in an effective way to explore the solution space in order to find better solutions and these techniques have been successfully applied to VRP and other combinatorial optimisation problems.

Metaheuristic techniques include Simulated Annealing, Tabu Search and Genetic Algorithm, amongst others. In particular, Genetic Algorithm (GA) is a stochastic search technique that closely mimics the metaphor of natural biological evolution. GA explores the problem domain by maintaining a population of individuals, which represents a set of potential solutions in the search space. The survival of each individual into the next generation is determined by its fitness, a performance measured based on an objective function that describes the problem. At each iteration, new individuals are created by selecting individuals according to their fitness and breeding them using genetic operators similar to natural genetics. The selection is carried out based on Darwin's principle of the survival of the fittest where stronger individuals are allowed to participate more in the reproduction of new individuals than the weaker ones, who may not even contribute at all. Using genetic operators, GA attempts to combine the good features found in each individual using a structured yet randomised information exchange in order to construct individuals which are better suited to their environment than the individuals that they were created from. Through the evolution of better and better individuals, it is hoped that the desired solution will be found.

Thangiah et al (1991) were among the first to introduce metaheuristics based on GA to solve VRPTW. This technique, which is known as GIDEON, has been successfully applied to VRPTW. Thangiah (1995) has modified the original GIDEON in order to improve the results. Recently, Thangiah et al (1995) extended the post-optimisation procedure to include the l-interchange mechanism. In order to diversify and intensify the search process, several strategies have been adopted from Tabu Search, Simulated Annealing and an *Oscillation Strategy* which allows the algorithm to iterate between feasible and infeasible search regions. A hybrid of Tabu Search and Simulated Annealing approaches has also been successfully implemented for VRPTW (Thangiah et al, 1995). Here, GA is only used as a mean of creating good initial solutions that will later be improved by some local optimisation techniques.

3. Description of the Problem

We first introduce some of the notation and abbreviations that are used below. Let $i, j = 1,\cdots, N$ and $k = 1,\cdots, K$ be the customers and vehicle indices where N and K denote, respectively, the total number of customers and the total number of vehicles available.

As in VRP, we assume that there are N

customers with known demand q_i, for $i = 1, \cdots, N$, to be serviced by a fleet of K homogeneous vehicles stationed at the depot $i = 0$. Each vehicle has a maximum capacity of Q_k, $k = 1, \cdots, K$, where Q_k is such that more than one customer may be assigned to a vehicle. Unlike VRP where we normally impose an upper bound on the number of vehicles to be used, here it is assumed that the number is unlimited and will be determined simultaneously with the routing and scheduling.

The service-time at each customer, i, involving pick-up or delivery of goods is denoted by $\boldsymbol{d}_i$ and it can only begin at time b_i within a time window defined by the earliest time e_i and the latest time l_i that a customer will permit the start of a service. Therefore, if a vehicle arrives before the beginning of the permitted service time, then the vehicle has to wait for a period w_i where $w_i = e_i - (b_j + \boldsymbol{d}_j + t_{ji})$, and $b_j + t_{ji}$ is the time the service is completed at customer j assuming that customer j precedes customer i. The variable t_{ji} is the time taken to travel from customer j to customer i and is normally assumed to be equal to the Euclidean distance d_{ji} between the two customers. The Euclidean distance is assumed to be symmetric, i.e. $d_{ji} = d_{ij}$. The beginning of service at customer i can therefore be explicitly expressed as $w_i = \max\{ e_i, b_j + \boldsymbol{d}_j + t_{ji} \}$.

In addition to the customer's time window, most formulations incorporate a scheduling horizon, which defines the working time of the respective vehicles by imposing a time window at the depot, denoted by e_0 and b_0.

Since VRPTW involves a time constraint, the solution to this problem consists of a set of directed arcs that must be followed. However if the time window constraints are very large, then part or all of the routes may be traversed in either direction without causing infeasibility. We would like to point out that for computational purposes, time, cost and distance are interchangeable.

4. GA-Based Approaches to VRPTW

We propose two methods: vertex sequencing and parallel savings approaches to demonstrate the application of GA in VRPTW. This procedure benefits the most from the hybridisation of GA and local search heuristics by incorporating an insertion heuristic proposed by Solomon (1987). On the other hand, the parallel savings approach allows us to show how a parallel route building algorithm based on GA can be adapted to VRPTW. Our second algorithm is similar to Blanton and Wainwright (1993), except for the criteria used in selecting the best route in which to insert the customer under consideration. We note that in both cases each chromosome encodes a list of permutation of cities/customers to be visited.

4.1 Vertex Sequencing for VRPTW

In this approach, each customer is assigned to the best vehicle (based on some criteria) following the sequence it appears on the chromosome. The best customer on each chromosome is used to initialise the first vehicle and subsequently, each unrouted customer is inserted in its best place in the emerging route using the insertion heuristic (Solomon (1987)). In the insertion heuristic a candidate is selected such that insertion in its best place yields an optimum value according to some criteria. This process involves two criteria that take into account the spatial and temporal aspects of the customers.

Let the sequence of the current route be represented as $(i_0, i_1, \ldots, i_m)$ where $i_0 = i_m = 0$ denote the depot. For each unrouted customer *u*, its best feasible insertion place with respect to capacity and time window constraints, in the emerging route, is computed as

$$a_1(i(u),u,j(u)) = \min_p \{a_1(i_{p-1},u,i_p)\} \qquad (1)$$

where $p = 1,\ldots,m$. Here, a_1 is defined as

$$a_1(i,u,j) = \mathbf{a}_1 a_{11}(i,u,j) + \mathbf{a}_2 a_{12}(i,u,j) \qquad (2)$$

where $\mathbf{a}_1 + \mathbf{a}_2 = 1$ and $\mathbf{a}_1, \mathbf{a}_2 \geq 0$. a_{11} and a_{12} are two factors based on the distance and time given by the expressions:

$$a_{11} = d_{iu} + d_{uj} - md_{ij}, \quad m \geq 0; \qquad (3)$$

$$a_{12} = b_{j_u} - b_j \qquad (4)$$

where m is a constant and b_{j_u} is the new time for the service to begin at customer *j*, given that *u* is on the route. Once the best place to insert each of the unrouted customers is identified, then the next step involves selecting the customer that will produce the best objective value. Therefore, a customer u^* is to be preferred for inclusion in the route when

$$a_2(i(u^*),u^*,j(u^*)) = \max_u \{a_2(i(u),u,j(u))\} \qquad (5)$$

where *u* is unrouted and feasible and $a_2(i,u,j)$ is such that

$$a_2(i,u,j) = l d_{u0} - a_1(i,u,j), \quad l \geq 0 \qquad (6)$$

where l is a constant.

In VRPTW, the insertion of new customers into a partial route always entails checking the feasibility with respect to the time window constraints of succeeding customers. An explicit testing of time feasibility at each customer can be carried out in *O*(*N*) time and this can consume a lot of computational time for reasonably large *N*. We implemented the procedure based on a *push forward* factor suggested by Solomon (1987). Here, we assume that each vehicle starts at the earliest possible time. Note that the departure time from the depot can be adjusted accordingly after the completion of the schedule to eliminate unnecessary waiting time. Let the sequence on the partially constructed feasible route be given as before, i.e. $(i_0, i_1, \ldots, i_m)$ and $i_0 = i_m = 0$ as depot. We denote $b_{i_p}^{new}$ as the new time the service at customer i_p begins, given that customer *u* is on the route. Also, let w_{i_r} be the waiting time at customer i_r for $p \leq r \leq m$. Assuming that the triangle inequality holds both for travel distances and times, the insertion of customer *u* defines *a push forward* in the schedule at i_p:

$$PF_{i_p} = b_{i_p}^{new} - b_{i_p} \geq 0 \qquad (7)$$

Hence,

$$PF_{i_r+1} = \max\{0, PF_{i_r} - w_{i_r}\}, \quad p \leq r \leq m; \qquad (8)$$

It is interesting to note that if $PF_{i_r} > 0$, some of the customers i_r, $p \leq r \leq m$ may become infeasible. Consequently, these customers should be examined sequentially for time feasibility until we find some customer, say i_r, $r < m$, for which $PF_{i_r} = 0$, or i_r is time infeasible. However, in the worst case, all the customers i_r, $p \leq r \leq m$ are examined. This result can be stated formally as:

Lemma 1

The necessary and sufficient conditions for time feasibility when inserting a customer, say u, between two adjacent customers i_{p-1} and i_p, $1 \le p \le m$, on a partially constructed feasible route $(i_0, i_1, i_2, \ldots, i_m)$, $i_0 = i_m = 0$ are

$$b_u \le l_u, \tag{9}$$

and

$$b_{i_r} + PF_{i_r} \le l_{i_r}, \quad p \le r \le m. \tag{10}$$

It should be noted that if non-Euclidean distance is used, it is possible that $PF_{i_p} < 0$, which results in all customers being time feasible. Also, since $i_m = 0$, Lemma 1 ensures that the vehicle will arrive at the depot within the scheduled time. However if inserting this customer violates one or all of the constraints, a new route is initiated using this customer. Since the number of vehicles is assumed to be unlimited, all the solutions obtained are feasible.

We note that following the clustering process, two customers that are adjacent to each other on the chromosome may not necessarily be adjacent on the routes. It is therefore necessary to rearrange the customers in the chromosome so that the chromosome represent the actual routing assigned by the insertion heuristic. This will allow GA to exploit the similarities in the chromosomes in order to search globally for an optimal sequence of customers to be visited.

In most route construction techniques, the choice of the first customer to be visited is a crucial factor in determining the quality of the final solution. Many procedures have been designed to find a suitable candidate to initialise each vehicle. Obviously, different criteria always result in different solutions, and in most cases the difference in the solutions can be quite significant. For GA-based algorithms, this problem does not arise since the first customer on the chromosome is always chosen to initialise the first vehicle. Subsequent vehicles are seeded using the first unrouted customer on the chromosome. Since GA works with a population of individuals, many alternative seeds can be exploited at the same time and the individuals that produce a competitive set of routes are then allowed to progress to the next generation.

4.2 A Time-Oriented Parallel Savings Method

In a time-oriented parallel savings method, the routes for all the vehicles are constructed simultaneously. This algorithm uses the first K genes to initialise each of the vehicles. With this approach, as in any other parallel route building procedure, the number of vehicles has to be determined *a priori*. Then, following the sequence on the chromosome, the cost of adding the first unrouted customer to the last customer on each vehicle is computed based on some selection criteria.

As pointed out by Solomon (1987), the savings heuristic alone may find it profitable to join two customers that are very close to each other geographically but far apart in terms of the earliest allowable service time. Such links will undoubtedly introduce extended periods of waiting time, which often results in a high opportunity cost since the vehicles can be serving other customers instead of waiting for the customer to start the service. Solomon (1987) proposed limiting the amount of waiting time when considering joining two customers. If joining two customers results in waiting time greater than a predefined value, then the two customers are not allowed to merge. However, we felt that it is more appropriate to use the weighted combination of the savings and the waiting time rather than defining an upper bound on the waiting time. Clearly, adding the customer to a vehicle, which produces maximum savings and minimum waiting time, is always preferred. Therefore, the selection criterion for each vehicle can be expressed mathematically as

$$a_1(k) = \mathbf{b}_1 a_{11} - \mathbf{b}_2 a_{12} \quad k = 1,2,\ldots,K \qquad (11)$$

where

$$a_{11} = \mathbf{a}_1 d_{ij} + \mathbf{a}_2 \max(0, e_{ij} - b_j) \qquad (12)$$

$a_{12} = s_{ij}$, $\mathbf{a}_1 + \mathbf{a}_2 = 1$ and s_{ij} is given as

(13)

$$s_{ij} = d_{0i} + d_{0j} - \mathbf{q}_1 d_{ij} + \mathbf{q}_2 \left| d_{0i} + d_{0j} \right|$$

(14)

with $\mathbf{q}_1 \in [1,3]$ and $\mathbf{q}_2 \in [0,1]$. Note that a_{11} accounts for the temporal aspects of the problem whilst a_{12} measures the savings obtained when this customer is merged with other customers on the route rather then serving them individually. Hence,

$$v^* = \arg\{\min_{k=1,\ldots,K} a_1(k)\} \qquad (15)$$

where v^* is the best vehicle to insert the current customer.

We should mention that, with time window constraints, as well as the capacity constraint, the feasibility of including the customer in each vehicle has to be validated first before any other criteria can be considered. This can be accomplished by computing the beginning of service at the customer if it is added on that particular route. If the time exceeds the latest allowable service time for that customer or the capacity of the vehicle is exceeded, then we associate an arbitrary large constant with the route to forbid the route from being selected. In addition, the time feasibility of the vehicle (ensuring that the addition of the new customer does not prevent the vehicle from reaching the depot within the scheduling horizon) must also be examined. In circumstances where no feasible route can be found, the customer is considered unserviced and the next customer on the sequence is then evaluated, unless all the customers have been examined. If the clustering process results in unserviced customers, the solution is considered infeasible and a penalty is imposed on the amount of unserviced demand. This ensures that infeasible solutions are ranked lower than feasible solutions. Since finding feasible solutions is often difficult in VRPTW, high penalties are normally selected so as to drive the search towards feasible region of the search space. Again, GA is used to search for a good sequence in the chromosome and a greedy heuristic based on a measure of saving and waiting time is used to find the best route for each customer. It is apparent from the way the clusterbuilder assigns customers to vehicles that a good chromosome should consist of customers with earlier deadlines at the beginning of the sequence and those with later ‘latest service times’ at the end of the string.

5. Objective Evaluation

The choice of a suitable objective function that mirrors the performance of a chromosome is essential in GA since its search is guided purely by the evaluation function. The algorithm requires that the objective value is able to discriminate between good and bad solutions. In most of the algorithms developed for VRPTW so far, a {lexicographic} ordering of parameters is normally employed. A solution that requires fewer vehicles is often preferred from the economic point of view, and this is followed by the total scheduling cost, total distance travelled and total waiting time, in order of decreasing priority. In algorithms with soft constraints, solutions with minimum tardiness are always favoured rather than those with smaller total scheduling cost, but with greater tardiness. It is often possible to reduce the total scheduling cost by increasing the number of vehicles, but most schedulers prefer a schedule that optimises the vehicle utilisation at the expense of an increase in total cost. Moreover, in some instances, it may be desirable to reduce the amount of {ideal} time (or waiting time) of each vehicle so that a more efficient schedule can be designed. Another important factor that may be of interest is the total distance covered by all the vehicles. Since an increase in the distance travelled inevitably results in an increase in operating cost, it may be desirable explicitly to express this factor in the objective to be optimised. However, empirical results have shown that reducing the total distance often results in an increase in the total waiting time and vice versa. This is because, as pointed out earlier, two customers that are close geographically may not necessarily be close from the temporal point of view. Therefore, placing more emphasis on the distance travelled will often result in unnecessary waiting time. It is clear that these solutions are such that improvement in any objective can only be achieved at the expense of degradation in other objective values. Since GA works with a population of individuals, it is possible to address this problem by exploiting the trade-off between the competing objective values.

In this study, several important factors, such as the total scheduling cost and the waiting time, are modelled using a weighted-sum approach. Here, all the relevant objectives are aggregated by combining them linearly with weights given to emphasise the relative importance of various objective functions. Let each objective function be represented as f_i, and

$$F(x) = \sum_{i=1}^{m} f_i f_i(x)$$

where $f_i \in [0,1]$ and $\sum_{i=1}^{m} f_i = 1$.

Different weights provide different set of solutions. Since we are only considering hard time window constraints, criteria that may be considered include minimising the number of vehicles, total scheduling cost, total distance covered, vehicle utilisation and total waiting time. As a preliminary investigation, minimising a combination of the total scheduling cost and the total amount of time spent waiting for the customers to start the service is considered. Although minimising the number of vehicles is equally important, we observed in all the test problems we tested that if the population contains solutions with fewer vehicles, the final best solution found after some termination criteria are attained always requires the same number of vehicles. It may, however, be useful to incorporate this factor in other type of problems. Moreover, this performance measure can easily be integrated in the objective function.

Thus, the objective function can formally be stated as

$$\min\{ f_1 F_1 + f_2 F_2 \} \qquad (16)$$

where F_1 and F_2 are, respectively, the normalised value of function

$$f_1 = \sum_{r} \sum_{(v_i, v_j) \in R_r} c_{ij}$$

and

$$f_2 = \sum_{i=1}^{N} w_i$$

where
$c_{ij} = d_{ij} + \boldsymbol{d}_i + w_j$ and $w_j = \max\{0, e_j - (b_i + \boldsymbol{d}_i + d_{ij})\}$
assuming that customer j precedes customer i, and $\boldsymbol{f}_1 + \boldsymbol{f}_2 = 1$.

For time-oriented parallel savings a slight modification has to be made to accommodate the infeasibility in the final solution. In order to avoid infeasible solutions being ranked higher than feasible solutions in the selection process, the number of unserviced customers is added to equation (16). The infeasible solutions are made more unattractive by multiplying the number of unserviced customers by a very large constant thus forcing the search towards feasibility. Hence,

$$\min\{\boldsymbol{f}_1 F_1 + \boldsymbol{f}_2 F_2 + gl\}$$

where l is the number of unserviced customers due to capacity or time window violations. The parameter g is normally chosen to be a sufficiently large value.

6. Development of the Test Problems

We have tested our algorithms on a set of data consisting of 30 customers taken from the 50-customer problem described in Christofides et al (1979). The location of these customers are generated randomly from a uniform distribution on a 100×100 Euclidean plane. The capacity of each vehicle is increased from 100 unit demand to 150 so as to allow more customers to be assigned to a vehicle. Furthermore, this ensures that the temporal aspect of the problem will dominate the search process for problems with short planning horizons. Using this problem we generate eight different problems that highlight several important aspects of VRPTW. The elements that may influence the performance of the algorithms include the tightness of the time window constraints (which indirectly determines the number of customers serviced by a vehicle), percentage of customers with time windows (sometimes known as the time window density) and the duration of the scheduling horizon. The tightness of the time windows, in this case, is given by the width of the time windows, measured by the difference between the latest l_i and the earliest e_i allowable service time. It is worth pointing out that, if the scheduling horizon is very large, the routing and scheduling cost is dominated by the capacity constraint. On the other hand the time window constraints determine the allocation of customers to vehicles in a problem with short planning horizon.

For each short and large scheduling horizon, we create problem sets with 25% and 100% of the customers with time window constraints. The time window for each customer is generated according to the method proposed in Solomon (1987). We consider two types of time windows. The first set consists of tight time windows with a mean of the time window width of 10 units time and the other is 100 units, which defines a wider time window. The characteristics of the problems investigated are given in Table 1.

Table 1: Characteristics of the test problems

Problem Number	Problem Size	Vehicle Capacity	Scheduling Horizon	% time window customers	Mean of the time window width
1	30	150	1000	25	10.00
2	30	150	1000	25	100.00
3	30	150	1000	100	10.00
4	30	150	1000	100	100.00
5	30	150	500	25	10.00
6	30	150	500	25	100.00
7	30	150	500	100	10.00
8	30	150	500	100	100.00

We assume that the time taken to service each customer is directly proportional to its demand.

7. Discussion and Results

The multi-objective vertex sequencing and time-oriented parallel GA were run on all eight data sets. For the vertex sequencing approach, each population consisted of 30 individuals whilst the time-oriented parallel GA consisted of 50 individuals per population. A smaller population was chosen for the vertex sequencing approach, which took longer to run because of the time-consuming insertion heuristic. All the programmes were terminated after 150 iterations. All other parameters such as the selective pressure (the bias towards the best individual), generation gap, crossover rate, mutation rate and insertion rate were fixed at $1.5, 0.8, 0.8, 0.33$ and 0.8, respectively. We used stochastic universal sampling to assign for each individual the expected number of offspring to be produced in the next generation. The fitness of each individual was assigned according to the non-linear ranking method (Chipperfield et al, 1993) and the reproduction strategy ensures that the least fit individuals are replaced by the new offspring. For the vertex sequencing approach we adapted the modified enhanced edge recombination operator (MEER) whereby the edge that connects the first and the last customers and those connecting customers on two separate vehicles are omitted in the construction of the edge list. The uniform order-based method was employed in the time-oriented parallel-savings GA. Scramble sub-list mutation was applied in both algorithms. Since different runs often produce different results, we simulated for each data set five times and the results tabulated in Table 2 were an average over the five runs. We note that the parameter values for the heuristic methods were chosen as recommended in Solomon (1987).

Table 2: Results for all the test problems

Problem number	Vertex Sequencing	Time-oriented parallel-GA	Heuristic 1[a]	Heuristic 2
1	1541 538.6 (22.6) 927.4 (3)	1875.6 783.1 (86.2) 1017.4 (4)	1938.4 709.9 1153.5 (4)	2245.2 1230.1 940.1 (4)
2	2473.6 1043.2 (19.8) 1355.4 (4)	7716.0 6446.6 (212.7) 1194.4 (13)	3081.4 1747.0 1259.4 (4)	3395.4 2004.6 1315.8 (4)
3	1551.0 553.5 (35.8) 922.5 (3)	2217.1 1122.0 (126.8) 1020.1 (4)	2288.0 1028.0 1185.0 (4)	2253.6 1184.1 994.5 (4)
4	2077.0 836.2 (44.0) 1165.7 (3)	2923.5 1803.1 (118.9) 1259.0 (8)	2636.1 1458.8 1102.3 (4)	2768.7 1377.5 1316.2 (4)
5	1192.0 86.6 (16.4) 1030.4 (3)	1613.2 518.6 (65.4) 1019.6 (4)	1489.5 144.0 1270.5 (4)	1700.3 209.5 1415.8 (4)
6	2280.3 825.1 (7.5) 1380.3 (6)	4849.4 3463.4 (231.6) 1386.0 (15)	2585.5 1303.4 1207.1 (7)	2706.7 1318.0 1313.7 (7)
7	1024.0 32.6 (17.8) 916.4 (3)	1336.9 227.9 (40.4) 1034.0 (4)	1230.3 265.4 889.9 (4)	1319.3 282.0 962.3 (4)
8	1373.7 141.0 (32.3) 1157.7 (3)	2556.5 1395.5 (115.2) 1086.0 (7)	1645.2 319.3 1250.9 (4)	1679.5 271.8 1332.7 (4)

[a] Solomon's insertion heuristics

For each heuristic, the best results tabulated in Table 2 were chosen according to the lexicographic ordering suggested in Solomon. The elements considered, in order of decreasing priorities, include the number of vehicles used, the scheduling cost, the total distance travelled and the total waiting time. A solution using fewer vehicles is always preferred regardless of the value of other parameters. If this results in a tie, then the solution with smaller total scheduling cost is chosen and so on. The first two columns of Table 3 display the average results for the vertex sequencing and time-oriented parallel GA. The other two columns show the best results obtained for the insertion (Heuristic 1) and time-oriented nearest neighbour (Heuristic 2) heuristics.

Interestingly the results depicted in Table 2 show that the vertex sequencing approach coupled with the insertion method produced results that are superior to those obtained using the time-oriented parallel savings method, the insertion heuristic and the time-oriented nearest neighbour method proposed by Solomon. It can be observed that the algorithm managed to find solutions with fewer vehicles in most of the test problems with the exception of Problem 2. The effect of tight time windows and short planning horizon in Problem 6 has resulted in more vehicles as shown by all the algorithms. In addition the vertex sequencing approach produces schedules with the least total scheduling cost and total waiting time for all the problems tested. In contrast the vertex sequencing approach performs slightly worse than the insertion heuristic with respect to the total distance travelled, only producing best overall total distance for three out of the eight problems. This is hardly surprising since the objective value is formulated so that greater emphasis is placed upon reducing the total scheduling cost and total waiting time. Note that reducing the total amount of waiting time often creates a schedule with a slightly higher total distance travelled.

Although the GA-based heuristics are more computationally expensive compared to the heuristics proposed by Solomon (Table 3), the improvement in the objective values obtained by the vertex sequencing method in all the problems tested are however quite substantial. We note that all the algorithms are run on Silicon Graphics on time sharing basis.

With most optimisation techniques, including those metaheuristics such as Simulated Annealing and Tabu Search, the final solution is confined to a single solution only. In contrast, GA-based procedures are able to offer several alternative solutions, which may be of interest to the users since it allows them to select the one that best approximates desired preferences at that particular time. For instance, it may be preferable to have a schedule with a slightly higher scheduling cost that distributes the

Table 3: Computational times (sec.) for the four methods

Problem Number	Vertex Sequencing	Time-oriented Parallel GA	Heuristic 1	Heuristic 2
1	1316.8	1743.8	4.0	1.0
2	950.5	2379.2	2.5	1.0
3	1463.3	102.4	4.4	1.0
4	963.9	861.2	3.3	1.1
5	1213.7	2046.2	3.8	1.1
6	871.3	2561.8	2.4	1.0
7	1316.8	1678.3	4.1	1.1
8	1002.9	2016.3	2.6	1.2

workload more evenly among all the vehicles, i.e. vehicle utilisation. Although a weighted-sum approach, as with any other aggregating procedure, results only in a single solution, other *good* solutions may be selected from those that lie within the vicinity of the optimal solution (Shaw and Fleming (1996)). Therefore these solutions, which may offer better schedules with respect to several other criteria that are not explicitly expressed in the objective function, may be selected to reflect the user's preferences at that time.

It is evident from Table 3 that the time-oriented parallel savings GA performs poorly, especially for problems with a high percentage of customers with time windows. Note that the algorithm requires an extremely large number of vehicles to schedule all the customers in those problem sets. Generally, it produces solutions that are worse then those obtained using the vertex sequencing approach. However, it produced slightly better solutions than the other two heuristics in problems with a large scheduling horizon and low time window density. When comparing these results, we have to bear in mind that the results reported for GA-based approaches are the average value as opposed to the best selected value from several runs (using different parameters) for the heuristic methods. When the constraints are tight, i.e. shorter planning horizon and high time window density, the time-oriented parallel savings method does not offer any advantage over the two heuristics developed by Solomon (1987).

In the time-oriented parallel savings approach, we observe that a difficulty occurs when the customers with early time deadlines are situated towards the end of the chromosome. Recall that the first K customers are used to initialise all the vehicles; since the customers are scanned according to the sequence in the chromosome, such situations often produces results that fail to schedule those customers. Similarly, if customers with later deadlines are situated at the beginning of the chromosomes, this often results in a bad schedule in which a larger number of vehicles may be required to route all the customers.

Preliminary experiments were carried out using a crossover based only on time precedence. Here genes at the same position in both parents are compared, and the one with earlier deadlines is selected to be placed in the offspring. We note that the time precedence crossover is taken from Blanton and Wainwright (1993). The algorithm was run using the same parameters as in the time-oriented parallel GA, with the exception of the selective pressure which we fixed at 1.19 as suggested by Blanton and Wainwright. Since heuristic operators, such as distance and time precedence crossovers, are susceptible to premature convergence due to the loss in diversity in the population, it is appropriate to reduce the bias factor (i.e. selective pressure) so as to minimise the effect of elitism. The average results for the time-oriented parallel GA using the uniform order-based and the time precedence operators, respectively, are tabulated in Table 4 which displays the total scheduling cost and the number of vehicles required by each algorithm in the final solution. All the results reported were averaged over five runs. It is worth mentioning that in cases where not all the five solutions produced were feasible, the results tabulated were taken as an average of all the feasible solutions. The number of vehicles for each run was determined by running the algorithms five times, and if no feasible solution was found the number of vehicles was increased accordingly.

Table 4 : Results for time-oriented parallel GA using various crossover operators

Problem Number	Uniform Order-based (UO)		Time Precedence (TP)	
1	1875.6	(4)	1883.7	(4)
2	7716.0	(13)	5701.3	(9)
3	2217.1	(4)	2803.1	(4)
4	2923.5	(8)	2947.5	(4)
5	1613.2	(4)	1616.6	(4)
6	4849.4	(15)	4676.3	(14)
7	1336.9	(4)	1348.9	(4)
8	2556.5	(7)	2176.7	(6)

It is observed that for problems with a high density of time windows, the algorithm using the time-precedence crossovers has dramatically reduced the number of vehicles used. This is most apparent in problems with long scheduling horizons but with a high density of customers with time windows (Problem 2 and 4). However, in cases where the scheduling horizon is small, the improvements are not so significant although a time-oriented GA using time precedence crossover performs slightly better than its counterpart. For Problems 1, 3, 5 and 7, the uniform order-based crossover produces better results than the time precedence. In general, the algorithm using the order-based operators took more computational time to converge than that using the time-precedence operator.

The performance of the vertex sequencing method was also evaluated on 6 out of the 56 problems compiled by Solomon[b] (1987). Problems R101, R102 and R201 consist of randomly generated customers while Problems C101, C102 consist of customers located in clusters. RC101 contains a mixture of randomly generated and clustered customers. Problems R101 and C101 have high time density and short scheduling horizon while Problem R201 have large vehicle capacities and large scheduling horizon.

Table 4 reports the best solution found over 5 runs using the vertex sequencing method, the best results from GIDEON and the recent results (the best) obtained by Rochat and Taillard (1995) using Tabu Search and the best solution they found using either Tabu Search or the diversification and intensification strategy. It is noted that our results are reported in the same manner as in Thangiah (1995) and Rochat and Taillard (1995). Both GIDEON and the methods proposed by Rochat and Taillard (1995) performed a post-optimisation on each of the solutions found. All the results reported in our algorithms have not benefited from post-optimisation procedure; obviously, further improvements could be obtained if post-optimisation is implemented.

The vertex sequencing method was run for a maximum of 350 generations using a population of 200 individuals. The best results found are tabulated in Table 5.

[b] The data sets were obtained via http://dmawww.epfl.ch/~rochat/rochat_data/

Table 5: Results for several benchmark problems

Problem Number	Vertex Sequencing	Solomon (1986)	GIDEON (1995)	Rochat & Taillard (1995)	Rochat & Taillard (1995)
R101	1674.2 (19)	1873 (21)	1770 (20)	1656.20 (19)	1650.80 (19)
R102	1589.6 (17)	1843 (19)	1549 (17)	1477.41 (18)	1486.12 (17)
R201	1447.9 (4)	1741 (4)	1478 (4)	1485.36 (4)	1281.58 (4)
C101	828.9(4) (10)	853 (10)	833 (10)	828.94 (10)	828.94 (10)
C102	839.3 (10)	968 (10)	832 (10)	828.94 (10)	828.94 (10)
RC101	1725.7 (15)	1867 (16)	1767 (15)	1737.03 (15)	1623.58 (15)

It is interesting to observe that the results reported for Problems R101, R201, C101 and RC101 are very competitive. In fact the distance of 828.94 is claimed to be the optimal value for Problem C101 when real distance is used (Rochat and Taillard (1995)). The vertex sequencing method has found a better solution than those reported by Thangiah (1995) and the Tabu Search by Rochat and Taillard (1995) for Problem R201 and RC101. It is noted that Problem R201 stresses more the routing rather than the scheduling since it has large vehicle capacities and scheduling horizon. Our method has found better (or comparable) solutions in 4 out of 6 solutions obtained by Thangiah and in 2 out of 6 by the Tabu Search method of Rochat and Taillard. However the GA-based algorithm was not able to improve on any of the best solutions reported by Rochat and Taillard. Generally, all our results are better than those obtained by Solomon (1987).

It should be mentioned that the vertex sequencing method uses a linear combination of the total distance travelled and the waiting time (with an exception of Problems R102 and RC101 rather than the lexicographic ordering employed in Thangiah and Rochat and Taillard. In Problems R102 and RC101 the objective function includes a third factor which emphasises the number of vehicles used. The objective function that did not include the number of vehicles often produces inferior solutions.

8. Conclusion

We have successfully shown how a local heuristic can be embedded inside Genetic Algorithm. The computational results obtained for the six benchmark problems tested are competitive when compared to those obtained by other metaheuristic methods. We believed that the computational time can be reduced further if the program were run on C++ rather than MATLAB.

References

Blanton, J.L. Jr., and Wainwright, R.L., 1993, *Multiple Vehicle Routing with Time and Capacity Constraints using Genetic Algorithms*, in S. Forrest, (ed.), Proceedings of the Fifth International Conference on Genetic Algorithm, Morgan Kaufmann Publisher, San Mateo, California, pp 452 -- 459.

Bodin, L.D., 1990, "Twenty Years of Routing and Scheduling", *Operations Research*, 38(4), pp 571 -- 579.

Chipperfield, A., Fleming, P., Pohlheim, H. and Fonseca, C., 1993, *Genetic Algorithm TOOLBOX for Use with MATLAB,* Department of Automatic Control and Systems Engineering, University of Sheffield, U.K.

Desrochers, M., Lenstra, J.K., Savelsbergh, M.W.P., and Soumis, F., 1988, "Vehicle Routing with Time Windows: Optimization and Approximation", in B.L. Golden and A.A. Assad, (eds.), Vehicle Routing: Methods and Studies, *Studies in Management Science and Systems*, North Holland, Amsterdam, 16, pp 65 -- 84.

Desrosiers, J., Dumas, Y., Solomon, M.M., and Soumis, F., 1995, "Time Constrained Routing and Scheduling}, in M.O". Ball, T.L. Magnanti, C.L. Monma and G.L. Nemhauser, (eds.), Network Routing, Handbook on Operations Research and Management Science, 8, North Holland, Amsterdam, pp 35 -- 139.

Koskosidis, Y., Powell, W., and Solomon, M., 1992, "An Optimization-based Heuristic for Vehicle Routing and Scheduling with Soft Time Window Constraints", *Transportation Science*, 26, pp 69 -- 85.

Potvin, J.-Y., and Rousseau, J.-M., 1995, "An Exchange Heuristic for Routeing Problems with Time Windows", *Journal of Operational Research Society,* 46 (12), pp 1433 -- 1446.

Pullen, H., and Webb, M., 1967, "A Computer Application to a Transport Scheduling Problem", *Journal of Computing*, 10, pp 10 -- 13.

Rochat, Y. and Taillard, E., 1995, "Probabilistic Diversification and Intensification in Local Search for Vehicle Routing", *Journal of Heuristics* 1, 1, pp 147 – 167.

Savelsbergh, M.W.P., 1984, "Local Search for Routing Problem with Time Windows", *Annals of Operations Research,* 4, pp 285 -- 305.

Shaw, K.J., and Fleming, P.J., 1996, "An Initial Study of Practical Multi-Objective Production Scheduling using Genetic Algorithm", *Proceedings of International Conference on Control*, 1, pp 479 -- 484.

Solomon, M.M., 1987, "Algorithm for the Vehicle Routing and Scheduling Problems with Time Window Constraints", *Operations Research*, 35 (2), pp 254 -- 265.

Solomon, M.M., and Desrosiers, J., 1988, "Time Window Constrained Routing and Scheduling Problem''s", *Transportation Science*, 22(1), pp 1 -- 13.

Thangiah, S.R., 1995, "Vehicle Routing with Time Windows using Genetic Algorithms", in L. Chambers, (ed), *Practical Handbook of Genetic Algorithms*: New Frontiers, II, pp 253 -- 277.

Thangiah, S.R., Nygard, K.E., and Juell, P.L., 1991, "GIDEON: A Genetic Algorithm System for Vehicle Routing with Time Windows", *Proceedings Seventh IEEE Conference on Artificial Intelligence Applications*, IEEE Computer Society press, Los Alamitos, California, pp 332 -- 325.

Thangiah, S.R., Osman, I.H., and Sun, T., 1995, "Metaheuristics for the Vehicle Routing Problems with Time Windows", *Research Report* No:UKC/IMS/OR94/8, Institute of Mathematics and Statistics, University of Kent, Canterbury.