```
    if (f[i]<=0) {
      path(i,j,NN);
      incr(j); }
  for (i=1;i<=NN;i++) bmatch[i]=f[i];
}
```

Revised 3/10/94.

AMERICAN JOURNAL OF MATHEMATICAL AND MANAGEMENT SCIENCES

ALGORITHMS FOR THE VEHICLE ROUTING PROBLEMS WITH TIME DEADLINES

Sam R. Thangiah
Artificial Intelligence and Robotics Laboratory, Computer Science Department
Slippery Rock University, Slippery Rock, PA 16057, U.S.A.

Ibrahim H. Osman
Institute of Mathematics and Statistics, University of Kent
Canterbury, Kent CT2 7NF, U.K.

Rajini Vinayagamoorthy and Tong Sun
Artificial Intelligence and Robotics Laboratory, Computer Science Department
Slippery Rock University, Slippery Rock, PA 16057, U.S.A.

SYNOPTIC ABSTRACT

The vehicle routing problem with time deadlines (VRPTD) is an extension of the classical vehicle routing problem (VRP) with constraints on the latest allowable time (deadline) for servicing each customer. The objective is to minimize the number of vehicles and the distance travelled without exceeding the capacity of the vehicles or violating the customer deadlines. VRPTD belongs to the class of NP-complete problems. As the computational time taken to solve such problems using exact methods is prohibitive, heuristic methods are used instead to obtain near optimal solutions for large-size problems. We develop three heuristics to solve the VRPTD: deadline sweep, push-forward insertion and genetic sectoring. The solutions obtained by these heuristics are improved using a local post-optimization procedure. Computational analysis of the three heuristics are reported on 25 problems consisting of 200 customers each with different geographical and temporal characteristics.

Key Words and Phrases: vehicle routing, time deadlines, genetic algorithms.

1993, VOL. 13, NOS. 3 & 4, 323-355
0196-6324/93/030323-33 $38.00

1. INTRODUCTION.

The vehicle routing problem with time deadlines (VRPTD) involves the design of a set of minimum cost delivery routes for a fleet of vehicles, originating and terminating at a central depot, that serves a set of customers with known demands and within specified time deadlines (latest allowable time) for accepting services. Each customer must be supplied by exactly one vehicle route. The total demand of any vehicle route must not exceed the maximum capacity of the vehicle. Each customer must be serviced within its specified time deadline. The VRPTD is the vehicle routing problem with time windows (earliest and latest arrival times), with the earliest arrival times removed [Solomon, 1987]. The VRPTD is also an extension of the standard vehicle routing problem (VRP) in which the time deadline constraints are added. Applications of routing and scheduling models arise in a wide range of practical decision making problems and efficient routing and scheduling of vehicles can potentially save public and private sectors millions of dollars per year. We refer to Osman (1993a) for the recent comprehensive survey in the literature on practical applications, implemented systems, development and classification of exact and approximate algorithms for more than ten different classes of routing and scheduling problems, after the early work of Bodin, Golden, Assad and Ball (1983). Special purpose surveys can be found in: Laporte (1992) and Osman (1993b) for the VRP; Solomon (1987), Golden and Assad (1988), Solomon and Desrosiers (1988), Desrochers, Desrociers and Solomon (1992) and Golden and Assad (1986) for VRP with time constraints.

Savelsbergh (1985) proved that obtaining a feasible solution to the travelling salesman problem with time windows (TSPTW) is, itself, a NP-complete problem. This demonstrates that the VRPTD is fundamentally more difficult than the TSPTW. Though optimal solutions to VRPTD problems can be obtained using exact methods, the computational time required to obtain such solutions is prohibitive. Heuristic methods often produce optimum or near-optimum solutions for large problems in a reasonable amount of computer time. Heuristic methods based on the Generalized Assignment method for solving VRPTD have been capable of solving small VRPTD problems with 25 customers and are not capable of obtaining feasible solutions for practical large-sized problems [Bolkan, 1986; Swenson, 1986; Nygard, Greenberg, Bolkan and Swenson, 1988]. Therefore, the development of heuristic algorithms that can solve large VRPTD in a reasonable amount of time is of primary interest.

In this paper we develop three heuristics for solving the VRPTD. The Deadline Sweep Heuristic (DSH) is a time oriented sweep approach that clusters customers and routes the vehicles within the clusters using a weighted cost function consisting of distance, time urgency and polar coordinate angle of a customer. The Push-Forward Insertion Heuristic (PFIH) uses an efficient insertion method to append customers to a vehicle route. The Genetic Sectoring Heuristic (GSH) is a cluster-first route-second heuristic that uses a Genetic Algorithm (GA) to sector the customers and the cheapest insertion method to route the vehicles within the sectors. The solutions obtained from these three heuristics are further improved using a local post-optimization procedure.

The local post-optimization procedure attempts to improve the solutions by allowing infeasible solutions to be accepted if the total travel time can be reduced even if the vehicles are overloaded, or if the vehicle is tardy in servicing the customers. A vehicle is considered to be overloaded if the total demand for all the customers to be visited by that vehicle is greater than its maximum capacity. A vehicle is considered to be tardy if the arrival time of the vehicle at a customer is after the customer's latest deadline. The three heuristics for solving the VRPTD were tested on 25 problems grouped into five data sets with 200 customers each. The five data sets are differentiated by the geographical placement of the customers. Each problem within the data set is differentiated by customer demands and time deadlines for servicing the customers.

Computational studies show that the GSH consistently obtains good feasible solutions for problems in which the customers have short time deadlines and/or are uniformly distributed. The DSH and the PFIH obtain better solutions than the GSH for problems in which the customers have long time deadlines and/or are tightly clustered.

Section 2 gives the description of the DSH for solving VRPTD. Sections 3 and 4 describe the PFIH and GSH methods. Section 5 describes the data sets used for comparing the performance of the three heuristics. Sections 6 and 7 report the computational results, and show the analysis for the above developed heuristics. Section 8 contains the summary and concluding remarks.

2. DEADLINE SWEEP HEURISTIC (DSH).

The Deadline Sweep Heuristic (DSH) is an extension of the Clarke-Wright and Gillet-Miller algorithms [Clarke and Wright, 1964; Gillett and Miller, 1974] for solving standard vehicle routing problems with time deadlines. The following notations will help in the description of the DSH, PFIH and GSH heuristics.

K	=	total number of vehicles.
N	=	total number of customers.
C_i	=	customer i, where i=1,...,N.
V_k	=	vehicle route k, where k=1,...,K.
O_k	=	total overload for vehicle route k, where k=1,...,K.
T_k	=	total tardiness for vehicle route k, where k=1,...,K.
D_k	=	total distance for a vehicle route k, where k=1,...,K.
d_{ij}	=	Euclidean distance (proportional to the travel time) from customer i to j, where i,j = 1,...,N.
d_0	=	the central depot.
l_i	=	latest time deadline at customer i, where i = 1,...,N.
t_i	=	total travel time to reach customer i, where i = 1,...,N.
u_{ij}	=	urgency of the customer j, i.e. $u_{ij} = l_j - (t_i + d_{ij})$, where i, j = 1,...,N.
p_i	=	polar coordinate angle of customer i, where i = 1,...,N.
s_i	=	pseudo polar coordinate angle of customer i, where i = 1,...,N.
F	=	fixed angle for Genetic Sectoring, $\mathrm{Max}[s_i,...,s_n]/2K$, where n = 1,...,N.
M	=	maximum offset of a sector in Genetic Sectoring, M= 3F.
B	=	length of the bit string in a chromosome representing an offset, B=5.
P	=	population size of the Genetic Algorithm, P=50.
G	=	number of generations the Genetic Algorithm is simulated, G=1000.
E_k	=	offset of the k^{th} sector, i.e, decimal value of the k^{th} bit string of size B, where k=1,...,K-1.
S_k	=	seed angle for sector k, where k=1,...,K-1.
S_0	=	initial seed angle for Genetic Sectoring, S_0=0.
α	=	weight factor for the distance.
β	=	weight factor for the urgency.
γ	=	weight factor for the polar coordinate angle.
η	=	penalty weight factor for an overloaded vehicle.
κ	=	penalty weight factor for the total tardy time in a vehicle route.

The DSH heuristic builds routes starting from the depot and adds a customer to the last customer in the current route using a cost function. The cost function takes into consideration the geographical and temporal characteristics of the customers. The heuristic search is executed on all customers in the problem that can be feasibly added to the current route without violating the time deadlines or capacity constraints. A new route is started any time the search fails. The heuristic search terminates when there are no more customers to be added. For a vehicle route with C_i as the last customer the DSH uses the following cost function to add an unrouted customer C_j to the route:

$$\text{cost from } C_i \text{ to } C_j = \alpha\,(t_i+d_{ij}) + \beta\, u_{ij} + \gamma\,((p_j/360)d_{ij}) \qquad (1)$$

This cost function (1) for DSH accounts for the geographical and temporal closeness of the customers, and is used to decide which customer should be added to the current route ended with customer C_i. We assume that there is an unlimited number of vehicles, K, which is large and determined by the heuristic to route all the customers. The weights for the cost function in (1) were derived empirically and were set as follows: α =0.7, β =0.2, and γ =0.1. The weights reflect the priority for the selection of the next customer in the order of distance, urgency and angle of the customer with respect to the last customer in the current route. When computing the polar coordinate angle of the next customer to be selected in (1), the angular value of the next customer is normalized in terms of the distance. This normalization allows comparison of distance, urgency and angular value in terms of a common unit. The flow of the DSH is described in Figure 1.

The DSH heuristic constructs routes by appending customers to the current route using the cost function (1) for selecting the next customer to be added. In order to obtain the first customer for a vehicle route, the cost function (1) is used to calculate the cost of all the customers from the depot. The unrouted customer with the lowest cost is chosen as the first customer to be added to the current route. The next customer is chosen by calculating the cost of all the unrouted customers from the customer last added to the current route and appending the customer with the least cost to the end of the current route.

When computing the cost of the next unrouted customer C_j, if C_j violates the time deadline or capacity constraint, then a penalty cost is added to prevent the consideration of C_j for addition into the current route. When no more customers can be added to the current route without violating the capacity or time constraints, a new route is initiated. The new tour starts from the unrouted customer nearest to the depot with the least cost and the process continues to add customers to the current route. The DSH method terminates when all customers are routed. The solution from the DSH method is improved using a local post-optimization procedure.

Step DSH-0: *Begin with an empty route starting from the depot.*
Set i=0, and r=1.
Step DSH-1: *If all customers are routed, go to Step DSH-6, otherwise, for all unrouted customers j: Compute the cost according to (1).*
Step DSH-2: *Sort the unrouted customers in a list in ascending order of their costs.*
Step DSH-3: *Select the first customer, j*, from the ordered list.*
If j can be appended after i without violating the capacity and time constraints,*
go to Step DSH-4,
otherwise, go to Step DSH-5.
Step DSH-4: *Append j* to the current route r.*
Update the capacity of the current route r.
Set i = j and go to Step DSH-1.*
Step DSH-5: *Begin a new route from the depot.*
Set r = r+1, and i=0.
Go to Step DSH-1.
Step DSH-6: *All customers are now routed.*
Either stop with the DSH solution, or go to Step DSH-7.
Step DSH-7: *Call the local post-optimization procedure.*
Stop the Deadline Sweep Heuristic with a local post-optimization (DSHO) solution.

FIGURE 1: Flow of the Deadline Sweep Heuristic (DSH).

The local post-optimization procedure improves a solution by shifting or exchanging customers between routes if it results in reduction of the total routing cost. The procedure shifts and exchanges customers between routes until no more improvements are found [Thompson and Psaraftis, 1988; Thangiah, 1991; Osman, 1991]. In the shift procedure, one customer is removed from a route and inserted into a different route. In the exchange procedure, one customer each from two different routes are removed and inserted into the other's route. In both shift and exchange procedures, improved solutions are accepted if the insertion results in the reduction of the total cost for routing the vehicles. The shift and exchange heuristics have theoretical properties [Osman and Christofides, 1994] and have been implemented successfully in many combinatorial problems [Osman, 1991, 1993b]. The local post-optimization procedure for the three heuristics described in this paper uses the shift and exchange of one and two customers between routes. For a detailed description of these procedures refer to [Osman, 1991].

3. PUSH-FORWARD INSERTION HEURISTIC (PFIH).

The Push-Forward Insertion method was introduced by Solomon for the VRP with time windows [Solomon, 1987]. It is an efficient method for computing the cost of inserting a new customer into the current route. The Push-Forward Insertion Heuristic (PFIH) is a modification of Solomon's Push-Forward Insertion method for the VRPTD.

FIGURE 2: The Push-Forward Insertion method for a set of customers in a vehicle route.

Figure 2 describes a route with m customers, with each customer having a deadline. The arrival time of a vehicle t_i at a customer is before its deadline l_i. For a customer C_i to be inserted between the depot and customer C_j, the insertion feasibility is checked by computing the amount of time that the arrival time of t_j is pushed forward. A change in the arrival time for t_j could affect the arrival time of all the successor customers of t_j in the current route. The insertion feasibility for C_i is computed by sequentially checking all the successor customers of C_i for feasibility. The Push-Forward for a customer C_j is 0 if the time propagated by the predecessor customers of C_j, because of the insertion of C_i into the route, does not affect the arrival time for t_j. The sequential checking for feasibility is continued until the Push-Forward for a customer is 0, a customer is pushed into being tardy, or, in the worst case, all customers are checked for feasibility.

The PFIH heuristic starts a new route by selecting an initial customer and then inserting customers into the current route until either the capacity of the vehicle is exceeded or it is not time feasible to add another customer to the emerging route. The criteria for selecting the first customer to be visited is based on the combined cost of the customer furthest away from the depot with the earliest deadline and the smallest polar coordinate angle.

Step PFIH-1: *Begin with an empty route starting from the depot.*
Set i=0, and r=1.

Step PFIH-2: *If all customers have been routed, go to step PFIH-8.*
For all unrouted customers j: Compute the cost according to (2), and sort them in ascending order of their costs.

Step PFIH-3: *Select the first customer, j*, from the ordered list with the least cost and feasible in terms of time and capacity constraints.*

Step PFIH-4: *Append j* to the current route r: set i = j*.*
Update the capacity of the route.

Step PFIH-5: *For all unrouted customers j: For all edges {k, l} in the current route, compute the cost of inserting each of the unrouted customers between k and l.*

Step PFIH-6: *Select an unrouted customer j* at edge {k*, j*} that has the least cost.*
If the insertion of customer j between k* and l* is feasible in terms of time and capacity constraints,*
insert customer j between k* and l*,*
update the capacity of the current route r, and
go to Step PFIH-5,
otherwise, go to Step PFIH-7.

Step PFIH-7: *Begin a new route from the depot.*
Set r = r + 1, and i = 0.
Go to Step PFIH-2.

Step PFIH-8: *All Customers have been routed.*
Either stop with a PFIH solution, or go to Step PFIH-9.

Step PFIH-9: *Call the local post-optimization procedure.*
Stop the Push-Forward Insertion Heuristic with a local post-optimization (PFIHO) solution.

FIGURE 3. Flow of the Push-Forward Insertion Heuristic(PFIH).

The cost function for selecting the first customer C_i is calculated using the following formula:

$$\text{Cost of } C_i = -\alpha\, d_{0i} + \beta\, l_i + \gamma\,((p_i/360)d_{0i}) \qquad (2)$$

The unrouted customer with the lowest cost is selected as the first customer to be visited. The weights for the three criteria were derived empirically and were set at $\alpha = 0.7$, $\beta = 0.1$, $\gamma = 0.3$. The priority for the selection of the customer is in order of distance, polar coordinate angle and latest time. When computing the polar coordinate angle of the customer with respect to the depot in (2), the angular value of the customer from the depot is normalized in terms of the distance. This

normalization allows comparison of the distance, latest deadline and angular value of the customer in terms of a common unit. Once the first customer is selected for the current route, the heuristic selects an unrouted customer C_l and an edge $\{i, j\}$ in the current tour that minimizes the total travel cost and inserts customer C_l between i and j if it does not violate time feasibility and capacity constraints. A new route is started when no more customers can be inserted into the current route without violating the capacity or time feasibility constraints. The flow of the PFIH is described in Figure 3.

4. GENETIC SECTORING HEURISTIC (GSH).

The Genetic Algorithm (GA) is an adaptive heuristic search method based on population genetics. The basic concepts of a GA were primarily developed by Holland [Holland, 1975]. Holland's study produced the beginnings of the theory of genetic adaptive search [DeJong, 1980; Grefenstette, 1986; Goldberg, 1989].

The GA is an iterative procedure that maintains a population of P candidate members over many simulated generations. The population members are string entities of artificial chromosomes. The chromosomes are fixed length strings with binary values (or alleles) at each position (or locus). Allele is the 0 or 1 value in the bit string, and the Loci is the position at which the 0 or 1 value is present in each location of the chromosome. Each chromosome has a fitness value associated with it (see Figure 4).

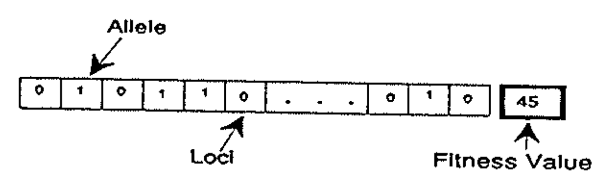

FIGURE 4: Description of a chromosome represented in a bit string format with the allele, loci and fitness value.

The chromosomes from one generation are selected for the next generation based on their fitness value. The fitness value of a chromosome is the payoff value that is associated with a chromosome. For searching other points in the search space, variation is introduced into the population chromosomes by using crossover and mutation genetic operators. Crossover is the most important genetic recombination operator. After the selection process, a randomly selected proportion of the chromosomes undergo a two point crossover operation and produce offspring for the next generation (see Figure 5).

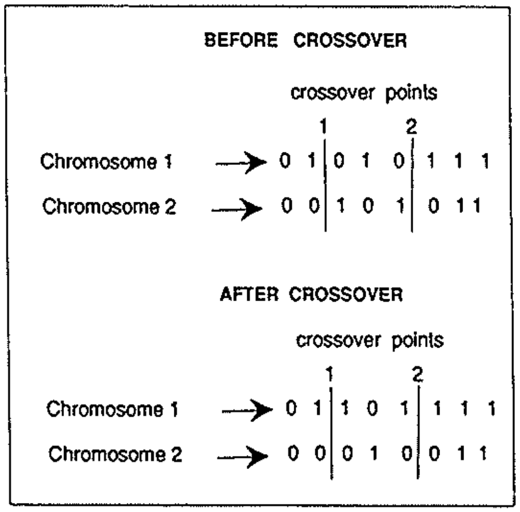

FIGURE 5: Example of a crossover operation with two chromosomes. The crossover points 1 and 2 are selected randomly.

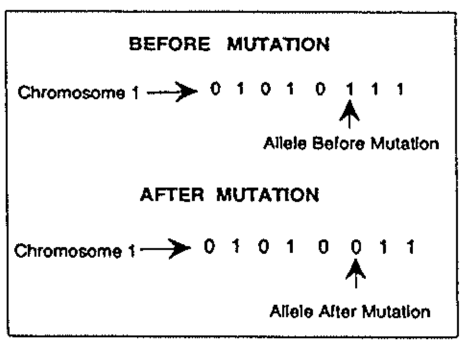

FIGURE 6: Example of a mutation operation on a chromosome. The location (locus) of the bit (allele) to be mutated is selected randomly.

Selection and crossover effectively search the problem space exploring and exploiting information present in the chromosome population by selecting and recombining primarily the offspring that have high fitness values. These two genetic operations generally produce a population of chromosomes with high performance characteristics. Mutation is a secondary operator that prevents premature loss of important information by randomly mutating alleles within a chromosome (see Figure 6). The adaptations in a GA are achieved by exploiting similarities present in the coding of the chromosomes. The termination criteria of a GA are convergence within a given tolerance or realization of the maximum number of generations to be simulated.

A clustering method using the GA has been highly successful in solving vehicle routing problems with time constraints, multiple depots and multiple commodities [Thangiah, 1993; Thangiah, Vinayagamoorthy and Gubbi, 1993; Thangiah and Nygard, 1993; Thangiah and Nygard, 1992a,1992b; Thangiah, Nygard and Juell, 1991]. In this paper we investigate the use of the genetic clustering method for solving VRPTD for a large number of customers.

The GA can be used to solve the VRPTD using the cluster-first route-second method. The GSH is a cluster-first route-second method. That is, given a set of customers and a central depot, the heuristic clusters the customers using the GA, and the customers within each sector are routed using the cheapest insertion method [Golden and Stewart, 1985].

The clustering of customers using a GA is referred to as Genetic Sectoring. As the sectors obtained by the GSH does not always result in a feasible solution, a local post-optimization procedure is used to improve the solution. The local post-optimization procedure shifts and exchanges customers between the routes to improve the solution obtained from the GA. The GSH heuristic allows exploration and exploitation of the search space to find good feasible solutions with the exploration being done by the GA and the exploitation by the local post-optimization procedure.

The GENESIS [Grefenstette, 1987] genetic algorithm software was used in the implementation of the GSH. The chromosomes in GENESIS are represented as bit strings. The sectors(clusters) for the VRPTD is obtained from a chromosome by subdividing it into K divisions of size B bits. Each subdivision is used to compute the size of a sector. The fitness value for the chromosome is the total cost of serving all the customers computed with respect to the sector divisions derived from it.

In an N customer problem with the origin at the depot, the GSH replaces the customer angles $p_1,...,p_N$ with pseudo polar coordinate angles. The pseudo polar coordinate angles are obtained by normalizing the angles between the customers so that the angular difference between any two adjacent customers is equal. This allows sector boundaries to fall freely between any pair of customers that have adjacent angles, whether the separation is small or large. The customers are divided into K sectors, where K is the number of vehicles, by planting a set of "seed" angles, $S_0,...,S_K$, in the search space and drawing a ray from the origin to each seed angle. The initial number of vehicles, K, required to service the customers is obtained using the DSH. The initial seed angle S_0 is assumed to be 0°. The first sector will lie between seed angles S_0 and S_1, the second sector will lie between seed angles S_1 and S_2, and so on. The Genetic Sectoring process assigns a customer, C_i, to a sector or vehicle route, V_k, based on the following equation:

$$C_i \textit{ is assigned to } V_k \quad \textit{if } S_k < s_i <= S_{k+1}, \textit{ where } k = 0,...,K-1.$$

Customer C_i is assigned to vehicle V_k if the pseudo polar coordinate angle s_i is greater than seed angle S_k but is less than or equal to seed angle S_{k+1}. Each seed angle is computed using a fixed angle and an offset from the fixed angle. The fixed angle, F, is the minimum angular value for a sector and assures that each sector gets represented in the Genetic Sectoring process. The fixed angle is computed by taking the maximum polar coordinate angle within the set of customers and dividing it by $2K$. The offset is the extra region from the fixed angle that allows the sector to encompass a larger or a smaller sector area.

The GA is used to search for the set of offsets that will result in the minimization of the total cost of routing the vehicles. The maximum offset, M, was set to three times the fixed angle to allow for large variations in the size of the sectors during the genetic search. If a fixed angle and its offset exceeds 360°, then that seed angle is set to 360° thereby allowing the Genetic Sectoring process to consider vehicles less than K to service all its customers. Therefore K, the initial

number of vehicles with which the GSH is invoked, serves as the upper bound on the number of vehicles that can be used for servicing all the customers.

The bit size representation of an offset in a chromosome, B, was derived empirically and was set at 5 bits. The decimal conversion of 5 bits results in a range of integer values between 0 and 31. The offsets are derived proportionately from the decimal conversion of the bit values using the decimal value 0 as a 0° offset and the bit value 31 as the maximum offset. Figure 7 describes the chromosome mapping used to obtain the offsets.

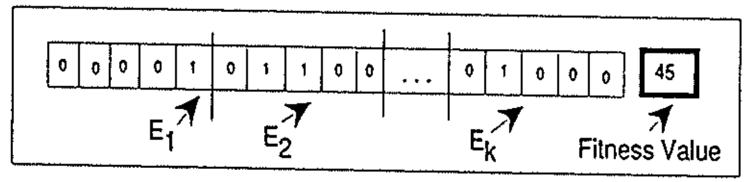

FIGURE 7: Representation of the offsets using a chromosome. Each offset is represented by five bits in the chromosome. The fitness value of the chromosome is the total route cost obtained using the offsets obtained from the chromosome.

The seed angles are derived from the chromosome using the following equation:

$$S_i = S_{i-1} + F + \left(3^{E_i\left(\frac{\log M}{\log 3}\right)}\right)\left(\frac{M}{2^B}\right) \tag{3}$$

The fitness value of a chromosome is the total cost of routing K vehicles for servicing N customers using the sectors formed from the set of seed angles derived from the chromosome. The seed angles are derived using the fixed angle and the offsets from the chromosomes. The formula (3) for calculating the seed angles uses an exponential function. The exponential function allows for large fluctuations in the seed angles with respect to the offsets derived from the chromosomes during the Genetic Sectoring process. The customers within the sectors, obtained from the chromosomes, are routed using the cheapest insertion method. The cheapest insertion method takes each unrouted customer in the sector and each edge $\{i, j\}$ in the current tour and computes the cost of inserting the unrouted customer between i and j. The unrouted customer that has the least insertion cost at edge $\{i, j\}$ is selected to be inserted between i and j.

The cost of inserting customer C_i into route V_k using the cheapest insertion method is calculated as follows:

$$\text{insertion cost of } C_i = D_k + \eta\, O_k + \kappa\, T_k \qquad (4)$$

The insertion cost formula (4) will accept infeasible solutions if the reduction in total distance is high enough to allow either a vehicle to be overloaded or be tardy. Overloading and tardiness in a vehicle route are penalized in the insertion cost function (4). When calculating the penalty weight factors η and κ for (4), η was set to ten percent of D_k and κ to one percent of D_k. The penalty values were chosen in this manner to allow penalization relative to the total distance travelled by the vehicle.

In the GSH each chromosome represents a set of offsets for a VRPTD. Therefore, a population of P chromosomes usually has P different solutions for a VRPTD. That is, there may be some chromosomes in the population that are not unique. At each generation P chromosomes are evaluated for fitness. The chromosomes that have the least cost will have a high probability of surviving into the next generation through the selection process. As the crossover operator exchanges a randomly selected portion of the bit string between the chromosomes, partial information about sector divisions for the VRPTD is exchanged between the chromosomes. New information is generated within the chromosomes by the mutation operator. The GSH uses selection, crossover and mutation to adaptively explore the search space for the set of sectors that will minimize the total cost of the routes over the simulated generations for the VRPTD. The GSH would utilize more computer time than either DSH or PFIH for obtaining a solution because every time the Genetic Sectoring Process is invoked it has to evaluate $P \cdot G$ vehicle routes, where P is the population size and G is number of generations to be simulated.

The parameter values for the number of generations, population size, crossover and mutation rates for the Genetic Sectoring process were derived empirically and were set at 1000, 50, 0.6 and 0.001. During the simulation of the generations, the GSH keeps track of the set of sectors obtained from the genetic search that has the lowest total route cost. The genetic search terminates either when it reaches the number of generations to be simulated or if all the chromosomes have the same fitness value. The best set of sectors obtained after the termination of the genetic search does not always result in a feasible solution. The infeasibility in a solution arises because of overloading or tardiness in a vehicle route. The solution obtained from the GA is improved using the local post-optimization procedure that shifts and exchanges customers between the vehicle routes.

The local post-optimization process is carried out until no more improvements can be made to the solution obtained from the GA. At the termination of the local post-optimization procedure, the customers are ranked in order of the sectors, and within the sectors in the sequence in which they are visited by the vehicles.

The customer angles, $p_1, \ldots, p_N$, are replaced with pseudo polar coordinate angles in order of the customer rank. The assignment of pseudo polar coordinate angles, using route and customer sequence, clusters together customers with geographical and temporal characteristics that can be serviced by a single vehicle. The customers with the new pseudo polar coordinate angles are once again used to form new sectors using the GA. The best set of sectors obtained from the GA are improved using the local post-optimization procedure. This iteration between Genetic Sectoring process and local post-optimization procedure is carried out a predetermined number of times and was set at 5. The flow of the GSH is described in Figure 8.

The Genetic Sectoring and local post-optimization procedures are symbiotic as the Genetic Sectoring is a meta-search strategy that forms the sectors and the local post-optimization procedure gives adjacency information about the customers back to the Genetic Sectoring process. These two procedures derive information from each other in order to obtain a good feasible solution.

Step GSH-1: *Set the number of cluster-route iterations: itermax = 5.*
Set the current iteration number: iter=0.
Set the bit string size for the offset: Bsize=5.

Step GSH-2: *Sort the customers in order of their polar coordinate angles, and assign pseudo polar coordinate angles to the customers.*
Set the lowest global route cost to infinity: g=∞.
Set the lowest local route cost to infinity: l=∞.

Step GSH-3: *Increment the number of iterations: iter=iter+1.*
If iter > itermax, go to Step GSH-7.

Step GSH-4: *If GA has terminated, go to Step GSH-5.*
For each chromosome in the population:
For each bit string of size BSize,
calculate the seed angle,
sector the customers, and
route the customers within the sectors using the cheapest insertion method.
If the cost of the current set of sectors is lower than l,
set l to the current route cost, and
save the set of sectors in lr.
If the cost of the current set of sectors is lower than g,
set g to the current route cost, and
save the set of sectors in gr.
Do Selection, Crossover and Mutation on the chromosomes.
Go to Step GSH-4.

Step GSH-5: *Do local post-optimization using the route lr.*
If no improvements can be made to route lr,
go to Step GSH-6.
If the current improved route has lower cost than l,
set l to the current cost, and
save the set of sectors in lr.
If the current improved route has lower cost than g,
set g to the current cost, and
save the set of sectors in gr.
Go to step GSH-5.

Step GSH-6: *Rank the customers of route lr in order of the sectors, and within the sectors in order of the sequence in which they are visited.*
Sort the customers by the rank.
Assign pseudo polar coordinates to the customers in order of the sorted rank.
Go to Step GSH-3.

Step GSH-7: *Stop the Genetic Sectoring Heuristic with a local post-optimization solution (GSHO) solution.*

FIGURE 8: Flow of the Genetic Sectoring Heuristic (GSH).

5. DATA SETS FOR VRPTD.

Generalized Assignment methods for solving VRPTD's have been capable of obtaining solutions for small VRPTD problems consisting of a maximum of 25 customers [Bolkan, 1986; Swenson, 1986; Nygard, Greenberg, Bolkan and Swenson, 1988]. Two problems solved by the Generalized Assignment method [Swenson, 1986] were also solved using the three heuristics and the results are listed in Table 1.

Table 1: Minimum distances obtained for the 25 customer problems requiring four vehicles using the GAM, DSH, PFIH and GSH heuristics.

Problems	GAM	DSH	DSHO	PFIH	PFIHO	GSHO
Problem 1	248	270	245	275	230	179
Problem 2	287{3}	358	290	263	244	220

Legend:

GAM: Generalized Assignment Method.
DSHO: Deadline Sweep Heuristic with local post-optimization.
PFIHO: Push-Forward Insertion Heuristic with local post-optimization.
GSHO: Genetic Sectoring Heuristic with local post-optimization.
DSH: Deadline Sweep Heuristic.
PFIH: Push-Forward Insertion Heuristic.
{}: Total tardy units.

The two 25 customer problems in Table 1 required 4 vehicles to service all the customers. The Generalized Assignment method obtained a feasible solution for the first problem but obtained an infeasible solution for the second problem. The other three heuristics obtained feasible solutions for both the problems. The Generalized Assignment method routes the vehicles without taking into consideration the time deadlines. Customer perturbations between routes are carried out to reduce the tardiness after the formation of the routes. As the solutions obtained by the Generalized Assignment method is not always feasible even for a small problems consisting of 25 customers, it would be impractical to use it for solving large-sized problems.

In order to solve realistic vehicle routing problems with time deadlines, five groups of problems were generated using the TNEWGEN program [Swenson, 1986]. Each data set was differentiated by the geographical placement of customers. The geographical characteristics range from uniformly distributed customers to customers who are clustered together over an area of 100 by 100 square units.

Each data set has five problems, and the characteristics of the customers within each data set highlight several factors that affect the behavior of routing and scheduling problems. These factors include geographical placement of customers, number of vehicles required for the problem, demand at each customer location, and the time deadline at each customer. Each problem in the data set consists of 200 customers. The demand for each customer ranged between 1 and 50 units. Each customer has a time deadline of either 100, 200, 300, 400, or 500 time units. The time deadlines are proportional to the distance travelled by the vehicle. The problems within each data set are differentiated by the probability distribution of the time deadlines for the customers. The distribution of time deadlines for the five problems within each data set is described in Table 2.

Table 2 the percentage distribution of the deadlines in time units for each of the problems in the data sets. For example, problem 2 of the data sets in Table 2 has 15 percent of the customers with a time deadline of 100 time units, 20 percent of the customers with a time deadline of 200 time units, 30 percent of the customers with a time deadline of 300 time units, 20 percent of the customers with a time deadline of 400 time units, and the rest of the customers with a time deadline of 500 time units. The complexity of the problem within each data set increases from problem 1 to problem 5 because of the number of customers with short time deadlines. That is, the greater the number of customers with short time deadlines, the more complex it is to obtain a feasible solution.

The data sets are labeled as RD, 1CD, 2CD, 3CD and 4CD. The depot for all the problems are at grid location (50, 50). The RD data set consists of uniformly distributed customers. The 1CD data set contains 1 cluster with the center of the cluster at grid point (20, 20). The cluster in data set 1CD contains 30 percent of the customers located within a radius of 10 miles of the cluster center. The 2CD data set contains 2 clusters with the cluster centers at grid points (20, 20) and (70, 70). Each cluster in data set 2CD contains 20 percent of the customers located within a radius of 10 miles of the cluster centers.

The data set 3CD contains 3 clusters with the cluster centers at grid points (20, 20), (20, 70), and (70, 70). Each cluster in data set 3CD contains 25 percent of the customers located within a radius of 10 miles of the cluster centers. The data set

4CD contains 4 clusters with the cluster centers at grid points (20, 20), (20, 70), (70, 20) and (70, 70). Each cluster in data set 4CD contains 25 percent of the customers located within a radius of 10 miles of the cluster centers. All the vehicles for the VRPTD begin at the central depot, and the vehicle fleet is assumed to be homogenous with a maximum capacity of 600 units.

Table 2: Percentage distribution of customer time deadlines for problems within each data set. For example problem 2 in each data set will have 15% of the customers with a time deadline of 100, 20% with a deadline of 200, 30% with a time deadline of 300, 20% with a time deadline of 400 and the rest with a time deadline of 500.

Deadline in time units	Prob. 1	Prob. 2	Prob. 3	Prob. 4	Prob. 5
100	20%	15%	10%	30%	60%
200	20%	20%	15%	30%	10%
300	20%	30%	50%	20%	10%
400	20%	20%	15%	10%	10%
500	20%	15%	10%	10%	10%

Legend:

Prob. 1 - Problem 1 for the data sets.
Prob. 2 - Problem 2 for the data sets.
Prob. 3 - Problem 3 for the data sets.
Prob. 4 - Problem 4 for the data sets.
Prob. 5 - Problem 5 for the data sets.

6. COMPUTATIONAL RESULTS.

The performance of the heuristics is based on the total distance travelled by the vehicles to service all the customers. The objective of the heuristics is to minimize the number of vehicles and distance required to service all the customers without violating the capacity or time deadline constraints.

The solutions for data sets RD, 1CD, 2CD, 3CD and 4CD were obtained using the DSH, PFIH and GSH methods. Table 3 tabulates the results obtained from the three heuristic methods. The DSH column in the table lists the solutions obtained by the Deadline Sweep Heuristic without local post-optimization, and DSHO with local post-optimization. The PFIH column lists the solutions obtained using the

Push-Forward Insertion Heuristic, and PFIHO with local post-optimization. The GSHO column is the Genetic Sectoring Heuristic with local post-optimization. In Table 3 the column "Prob" is the problem number, "Dist" is the total distance travelled by the vehicles, the values within the square brackets are the number of vehicles required for the solution, the values in the curly brackets are the total tardy units for the solution and "CPU" is the total CPU time taken on a NeXT 68040(25MHz) computer to get a solution. In Table 3 the best feasible solution obtained in terms of the minimum number of vehicles and distance between the three heuristics for a problem is in bold followed by an asterik.

The GSHO did better for all the problems in comparison to the solutions obtained from DSH and PFIH that were locally post-optimized using only feasible moves. When infeasible moves were allowed during local post-optimization of the solutions found by DSH and PFIH, it resulted in some solutions obtained by DSH and PFIH attaining better solutions than those of the GSHO. The locally post-optimized DSHO and PFIHO solutions can become infeasible as the insertion cost function (4) used in the local post-optimization procedure allows acceptance of infeasible solutions. The GSHO obtained feasible solutions for all the problems. The DSHO, PFIHO and GSHO used the same insertion-cost function (4) for local post-optimization of its solutions. The code for the heuristics were written in the C language and executed on the NeXT computer system.

7. COMPUTATIONAL ANALYSIS

The solutions obtained from DSHO, PFIHO and GSHO were compared with each other. Only the locally post-optimized solutions of the three heuristics were compared as local post-optimization is an integral part of the GSH. As evident from Table 3, for problems in which a large percentage of the customers have short time deadlines, specifically problem 5 of all the data sets except 4CD, the GSHO consistently obtained feasible solutions while DSHO and PFIHO obtained infeasible solutions. Out of the 25 problems solved using the three heuristics, DSHO obtained 3 best feasible solutions, PFIH 9 best feasible solutions and GSHO 14 best feasible solutions.

Table 3: Results from the DSH, PFIH and GSH methods with and without local post-optimization for the data sets RD, 1CD, 2CD, 3CD and 4CD.

Prob	DSH		DSHO		PFIH		PFIHO		GSHO	
	Dist	CPU	Dist	CPU	Dist	CPU	Dist	CPU	Dist	CPU
RD1	3770[9]	0.3	2011 [9]	10	2674[10]	1	1746[9]	23	**1736[9]***	32
RD2	3549[9]	0.3	1854 [9]	10	2555[10]	1	1762[9]	24	**1623[9]***	34
RD3	2888[9]	0.2	1838 [9]	18	2211[9]	1	1634[9]	18	**1579[9]***	39
RD4	3272[9]	0.3	1872 [9]	9.	2969[12]	1	1702[9] {15}	21	**1572[9]***	34
RD5	4002[8]	0.4	2012 [8]{120}	7	2939[15]	1	1915[10]{86}	102	**1770[8]***	37
1CD1	3002[9]	0.3	1699 [9]	13	2234[9]	1	**1449[9]***	73	1796[9]	50
1CD2	3059[9]	0.4	**1507 [9]***	8	2421[10]	1	1689[9]	75	1648[9]	49
1CD3	3229[9]	0.3	1747 [9]	234	2249[9]	1	1550[9] {6}	53	**1745[9]***	46
1CD4	3192[9]	0.3	1711 [9]	10	2551[11]	2	1705[9] {25}	70	**1687[9]***	42
1CD5	3508[8]	0.3	1725 [8]	325	2955[15]	0.3	1635[10] {30}	73	**1459[8]***	37
2CD1	3330[9]	0.4	1561 [9]	15	2351[10]	0.4	1436[9]{18}	71	**1490[9]***	77
2CD2	3051[9]	0.4	1629 [9]	8	2204[10]	0.4	**1508[9]***	29	1613[9]	47
2CD3	2931[9]	0.3	1710 [9]	28	2144[9]	0.4	1514[9] {9}	18	**1566[9]***	54
2CD4	2172[9]	0.3	1548 [9]	396	2426[12]	0.3	1695[10] {71}	24	**1538[9]***	37
2CD5	2441[9]	0.3	1605 [9] {49}	650	2453[13]	0.3	1586[10] {10}	16	**1595[9]***	49
3CD1	2712[9]	0.3	1400[9]	21	2158[10]	0.2	**1286[9]***	16	1375[9]	84
3CD2	2750[9]	0.4	1507[9]	10	1897[9]	0.3	**1344[9]***	15	1354[9]	44
3CD3	2524[9]	0.3	1129[9]	21	1800[9]	0.3	**1081[9]***	17	1376[9]	53
3CD4	2644[9]	0.3	1513[9]	11	2183[9]	0.4	1377[8] {3}	16	**1426[8]***	59
3CD5	2251[9]	0.3	1370[9] {11}	218	2642[13]	0.4	1533[9] {5}	27	**1490[9]***	45
4CD1	1969[9]	0.3	**1157[9]***	14	1585[9]	0.3	1182[9]	13	1280[9]	73
4CD2	1924[9]	0.4	978[9]	10	1441[9]	0.3	**916[9]***	10	1211[9]	68
4CD3	1610[9]	0.4	1187[9]	160	1578[9]	0.4	**1036[9]***	16	1289[9]	39
4CD4	2030[9]	0.3	1193[9]	22	1663[10]	0.4	**1038[9]***	16	1396[9]	96
4CD5	1985[9]	0.3	1115[9]	7	1852[9]	0.4	**968[9]***	10	1139[9]	42

Legend:

Prob: The problems for the data set.
DSHO: Deadline Sweep Heuristic with local post-optimization.
PFIH: Push-Forward Insertion Heuristic.
PFIHO: Push-Forward Insertion Heuristic with local post-optimization.
GSHO: Genetic Sectoring Heuristic with local post-optimization.
*: Best feasible solution for the problem.

DSH: Deadline Sweep Heuristic.
Dist: Total Distance.
CPU: CPU time in seconds on a NeXT computer.
{ }: Total tardy units
[ ]: Number of vehicles.

The solutions obtained by DSH for problems 4 and 5 in the data sets were better than the solutions obtained by PFIH in terms of the number of vehicles required to visit all the customers. The GSHO obtained better solutions than either DSHO or PFIHO for all the problems in data set RD in which the customers are uniformly distributed. Both the DSHO and the PFIHO obtain better solutions than the GSHO for all the problems in data set 4CD which consists of tightly clustered customers. The GSHO does better than both DSHO and PFIHO for some of the problems in data set 1CD.

Table 4 gives the mean and standard deviations for the solutions reported in Table 3. The best solutions with the minimum average distance, obtained by one of the three heuristics for the data sets, are in bold followed by an asterik. The best average solutions shown in Table 3 do not take into consideration the infeasible solutions obtained by the heuristics for some of the problems in the data set. For data set RD, GSHO outperforms both DSHO and PFIHO. Though PFIHO obtains better solutions than DSHO for data sets 2CD and 3CD, some of the solutions obtained by PFIHO for problems in those data sets are infeasible. Both DSHO and PFIHO outperform the GSHO heuristic for data set 4CD. The solutions obtained by DSHO and PFIHO for data set 4CD have no infeasible solutions.

The mean performances of the three heuristics indicate that though GSHO does uniformly better than DSHO on the data set RD, and DSHO and PFIHO do better than GSHO on data sets 1CD, 2CD, 3CD and 4CD. The solutions indicate that the GSHO heuristic does well for problems with few clusters and the DSHO and PFIHO for problems with many clusters. GSHO consistently attains feasible solutions for all the problems in the data sets. The variance in the CPU time required by the DSHO to obtain a solution is high compared to that of PFIHO and GSHO.

Figure 9 illustrates the average distance obtained by DSHO, PFIHO and GSHO for the five data sets. The GSHO obtains the best average solution for data set RD. The PFIHO obtains the best average solutions for data sets 1CD, 2CD, 3CD and 4CD. It should be noted that for problems in the data sets with time deadlines the PFIHO tends to obtain infeasible solutions, and the GSHO obtains feasible solutions for all problems with tight time deadlines.

Table 4: Mean and standard deviation of the solutions obtained by DSHO, PFIHO and GSHO for the data sets RD, 1CD, 2CD, 3CD and 4CD.

Data Set.		DSHO			PFIHO			GSHO	
		Dist	Tar	CPU	Dist	Tar	CPU	Dist	CPU
RD	Mean	1917.4 [8.8]	24	11.0	1751.8 [9.2]	20.2	37.8	**1656.9*** [8.8]	35.2
	S.D	86.7 [0.5]	53.7	4.1	103.84 [0.45]	37.5	36.2	91.5 [0.5]	2.9
1CD	Mean	1671.8 [8.8]	0	118.1	**1605.6*** [9.2]	12.2	67.9	1667.0 [8.5]	44.9
	S.D	96.4 [0.5]	0	150.8	106.45 [0.45]	14.3	10.5	129.2 [9.8]	5.2
2CD	Mean	1610.6 [9]	10.8	219.4	**1547.8*** [9.2]	21.6	31.5	1560.4 [9.0]	53.1
	S.D	64.5 [0]	21.3 9	291.4	97.92 [0.55]	28.3	22.4	48.61 [0]	14.8
3CD	Mean	1383.8 [9]	2.2	56.2	**1324.7*** [8.8]	1.6	18.2	1404 [9.0]	57.3
	S.D	155.9 [0]	4.92	90.6	163.9 [0.45]	2.3	4.9	54.8 [0]	15.9
4CD	Mean	1126 [9]	0	42.5	**1028*** [9]	0.2	13.3	1263.0 [9]	63.3
	S.D	88.3 [0]	0	66.2	100 [0]	0.44	3.6	95.8 [0]	23.6

Legend:

Data Set: The data sets for the VRPTD
DSHO: Deadline Sweep Heuristic with local-post-optimization.
PFIHO: Push-Forward Insertion Heuristic with local post-optimization.
GSHO: Genetic Sectoring Heuristic with local post-optimization.
Mean: Mean value.
*: Best average feasible/infeasible solution for the data set.

Dist: Total distance.
CPU: CPU time in seconds on a NeXT machine.
Tar: Total tardy units.
[]: Number of vehicles.
S.D: Standard Deviation value.

In Figure 9 the PFIHO outperforms the DSHO for all the problems in the data sets. The GSHO heuristic does significantly better than DSHO for data sets RD, 1CD and 2CD. The DSHO does significantly better than GSHO for data sets 3CD and 4CD. GSHO obtains better solutions than either DSHO or PFIHO for vehicle routing problems with time deadlines has customers uniformly distributed and/or short time deadlines. DSHO and PFIHO do well in solving problems that are tightly clustered and/or with long customer deadlines

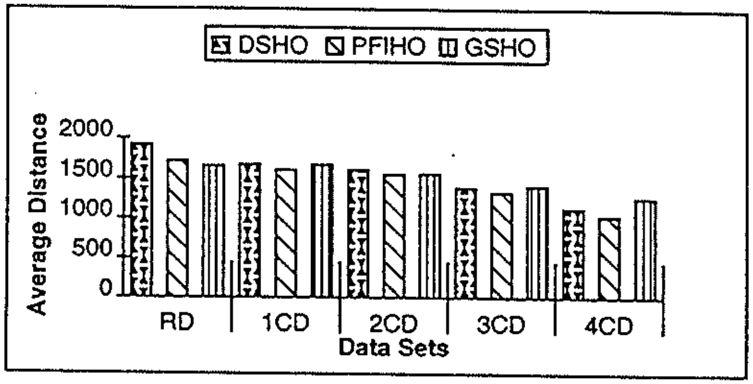

FIGURE 9: Plot of the average distances obtained by DSHO, PFIHO and GSHO heuristics for the RD, 1CD, 2CD, 3CD and 4CD data sets.

The Wilcoxon rank signed test is a non-parametric statistical test used for the statistical analysis of observations that are paired and can be applied to compare the performance of heuristics [Golden and Steward, 1985]. The Wilcoxon signed rank could not be used to analyze the significance of the solutions obtained by the DSHO, PFIHO and GSHO heuristics on the five data sets as it uses signed ranks of differences to assess the difference in two locations of the two populations. For example for data set 2CD the standard deviation ratio of populations PFIHO and GSHO is 2.01. Thus the standard deviations differ by a factor of 2 and the variances by a factor of 4. Let us apply the Wilcoxon rank signed test to compare the performance of the two heuristics PFIHO and GSHO for data set 1CD with the null hypothesis being E[PFIHO]=E[GSHO]. A one-sided test with the alternate hypothesis E[PFIHO]<E[GSHO] would yield -3 as the sum of the weighted ranks for the two heuristics. The critical region for the test with $\alpha = 0.05$ and $n=5$ would indicate that only in one out of twenty trials would the weighted sum of the ranks for the two heuristics exceed 1. As the sum of the weighted ranks is -3, the null hypothesis is rejected and we conclude that PFIHO performs better than GSHO for the data set 1CD. The solutions obtained by PFIHO and GSHO data set 1CD in Table 2 indicate that GSHO obtains better solutions than PFIHO for four out of the five problems which is in contradiction of the Wilcoxon signed rank test.

As the Wilcoxon test assumes only a difference in location and not in standard deviation, it is not a suitable for comparing the performance of the three heuristics. This difference in population arises as some of the solutions obtained by the DSHO and PFIHO heuristics are infeasible and are compared with the feasible solutions obtained by GSHO. The GSHO heuristic performs better in all the problems with respect to DSHO and PFIH, if the post-optimization process for the DSHO and the PFIHO considered only moves that were feasible. The DSHO and PFIHO with local post-optimization procedures were allowed infeasible moves, which could result in an infeasible solution, in the local post-optimization procedure for obtaining solutions that are competitive with GSHO.

8. SUMMARY AND CONCLUSION

In this paper we introduced three heuristics, a time oriented Deadline Sweep Heuristic (DSH), a Push-Forward Insertion Heuristic(PFIH) and a Genetic Sectoring Heuristic (GSH) for solving vehicle routing problems with time deadlines. The Deadline Sweep Heuristic (DSH) is an extension of the Clarke-Wright and Gillet-Miller algorithms [Clarke and Wright, 1964; Gillett and Miller, 1974] for solving standard vehicle routing problems with time deadlines. The Push-Forward Insertion Heuristic uses a method similar to the one described in [Solomon, 1987]. The Genetic Sectoring Heuristic uses a cluster-first route-second method. The Genetic Sectoring Heuristic uses the genetic algorithm to form customer clusters and routes the customers within the clusters using the cheapest insertion method. The three heuristics were used to solve problems consisting of 200 customers that varied in geographical distribution of customers, demands and time deadlines. The solutions obtained from the three heuristics are improved using a local post-optimization procedure that shifts or exchanges customers between the routes if it leads to a reduction in the total route cost. The local post-optimization procedure allowed acceptance of infeasible solutions to minimize the total route cost. The computational analysis of the solution shows that the Genetic Sectoring Heuristic does well for problems in which the customers are distributed uniformly and/or with short time deadlines. The Deadline Sweep Heuristic and the Push-Forward Insertion Heuristic do well for problems in which the customers are tightly clustered and/or have long time deadlines.

ACKNOWLEDGEMENTS

The authors wish to thank the three referees and the editors (Prof. Bruce Golden and Prof. Edward Dudewicz) for valuable criticisms and suggested improvements. This material is based upon work partly supported by the Slippery Rock University Faculty Development Grant No. FD2E-030 and the National Science Foundation Grant No. USE-9250435. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and not necessarily reflect the views of Slippery Rock University or the National Science Foundation.

REFERENCES

Bodin, L., B. Golden, A. Assad and M. Ball (1983). The State of the Art in the Routing and Scheduling of Vehicles and Crews. Computers and Operations Research 10, 63-211.

Bolkan, Waverly E. (1986). Algorithm for the Vehicle Routing Problem with Deadlines. Masters Thesis, Department of Computer Science and Operations Research, North Dakota State University, Fargo, North Dakota.

Clarke, G. and J. W. Wright (1964). Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. Operations Research 12, 568-581.

DeJong, Kenneth (1980). Adaptive System Design: A Genetic Approach. IEEE Transactions on Systems, Man and Cybernetics 10 (9), 566-574.

Desrochers, M., J. Desrociers and M. Solomon (1992). A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows. Operations Research 40(2), 342-354.

Gillett, B. and L. Miller (1974). A Heuristic Algorithm for the Vehicle Dispatching Problem. Operations Research 22, 340-349.

Goldberg D. E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Publishing Company, Inc., New York.

Golden B. and A. Assad (eds.) (1988). Vehicle Routing: Methods and Studies. North Holland, Amsterdam, 1988.

Golden, B and A. Assad (eds.) (1986). Vehicle Routing with Time Window Constraints: Algorithmic Solutions. American Journal of Mathematical and Management Sciences 15, American Sciences Press, Columbus, Ohio.

Golden, B. and W. Stewart (1985). Empirical Analysis of Heuristics, in the Traveling Salesman Problem, (E. Lawler, J. Lenstra, A. Rinnoony Kan and D. Shmoys, Eds.). Wiley-Interscience, New York, 207-249.

Grefenstette, John J. (1987). A Users Guide to GENESIS. Navy Center for Applied Research in Artificial Intelligence, Naval Research Laboratory, Washington D. C. 20375-5000.

Grefenstette, John J. (1986). Optimization of Control Parameters for Genetic Algorithms. IEEE Transactions on Systems, Man and Cybernetics, 16(1), 122-128.

Holland, J. H. (1975). Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, Michigan.

Laporte, G. (1992). The Vehicle Routing Problem: An Overview of Exact and Approximate Algorithms. European Journal of Operational Research, 59, 345-358.

Nygard, K. E., P. Greenberg, W. E. Bolkan, and J. E. Swenson (1988). Generalized Assignment Methods for the Deadline Vehicle Routing Problem. In Vehicle Routing: Methods and Studies, (Golden, B.L., and A.A. Assad, Eds.), North Holland, Amsterdam, 107- 125.

Osman, I. H. and N. Christofides (1994). Capacitated Clustering Problems by Hybrid Simulated Annealing and Tabu Search. International Transactions in Operational Research, Forthcoming.

Osman, I. H. (1993a). Vehicle Routing and Scheduling: Applications, Algorithms and Developments. Proceedings of the International Conference on Industrial Logistics, Rennes, France.

Osman, I. H. (1993b). Metastrategy Simulated Annealing and Tabu Search Algorithms for the Vehicle Routing Problems. Annals of Operations Research, 41, 421-451.

Osman I. H. (1991). Metastrategy Simulated Annealing and Tabu Search Algorithms for Combinatorial Optimization Problems. Ph.D thesis, The Management School, Imperial College, University of London.

Savelsbergh, M. W. P. (1985). Local Search for the Vehicle Routing and Scheduling Problem with Time Windows. Annals of Operations Research, 4, 285-305.

Solomon, M. M., and J. Desrosiers (1988). Time Window Constrained Routing and Scheduling Problems. Transportation Science, 22, 1-13.

Solomon, M. M. (1987). Algorithms for the Vehicle and Scheduling Problems with Time Window Constraints. Operations Research, 5, 254-265.

Swenson, E. J. (1986). The Vehicle Routing Problem with Time Constraints. Masters Thesis, Department of Computer Science and Operations Research, North Dakota State University, Fargo, North Dakota.

Thangiah, Sam R. (1993). Vehicle Routing with Time Windows using Genetic Algorithms. Technical Report SRU-CpSc-TR-93-22, Artificial Intelligence and Robotics Laboratory, Computer Science Department, Slippery Rock University, Slippery Rock, PA 16057.

Thangiah, Sam R., R. Vinayagamoorthy and A. Gubbi (1993). Vehicle Routing with Time Deadlines using Genetic and Local Algorithms. Proceedings of the Fifth International Conference on Genetic Algorithms, Urbana-Champagne, Illinois, 506-513.

Thangiah, Sam R. and K. Nygard (1993). Dynamic Trajectory Routing using an Adaptive Search Strategy. Association for Computing Machinery's Symposium on Applied Computing, Indianapolis, Indiana. 131-138.

Thangiah, Sam R. and K. Nygard (1992a). School Bus Routing using Genetic Algorithms. Proceedings of the Applications of Artificial Intelligence X: Knowledge Based Systems, Orlando, 387-397.

Thangiah, Sam R. and K. Nygard (1992b). MICAH: A Genetic Algorithm System for Multi-Commodity Networks. Proceedings of the Eight IEEE Conference on Applications of Artificial Intelligence, Monterrey, California, 240-246.

Thangiah, Sam R., K. Nygard and P. Juell (1991). GIDEON: A Genetic Algorithm System for Vehicle Routing Problems with Time Windows. Proceedings of the Seventh IEEE Conference on Artificial Intelligence Applications, Miami, Florida, 322-328.

Thompson, P. and H. Psaraftis (1993). Cyclic Transfer Algorithms for Multivehicle Routing and Scheduling Problems. Operations Research, 41, 935-946.

APPENDIX A: SAMPLE PROBLEM FROM THE DATA SET

In this appendix one sample problem from the data sets is presented. The data sets for the VRPTD and the code can be obtained by writing to the first author.

PROBLEM: RD1

x distance	y distance	
100	100	
depot x	depot y	number of nodes
50	50	200

x	y	demand	node	theta	distance	deadline
76	5	23	1	0.24	712	500
28	5	37	2	0.73	235	500
95	6	37	3	0.82	908	300
90	6	11	4	0.87	857	200
98	7	28	5	1.35	936	200
92	8	10	6	2.23	873	200
68	8	19	7	3.17	631	400
96	11	2	8	4.26	915	500
67	9	26	9	4.39	626	400
19	6	32	10	4.86	141	400
98	13	45	11	4.94	940	300
44	8	26	12	5.64	396	400
87	13	48	13	6.10	827	500
20	6	26	14	6.81	160	500
43	9	35	15	7.18	384	400
55	11	35	16	7.25	507	100
44	10	28	17	7.28	402	400
25	8	9	18	8.75	210	100
58	13	27	19	9.31	544	300
75	17	13	20	9.77	713	100
87	20	16	21	10.35	840	300
85	20	23	22	10.50	822	300
58	15	30	23	10.72	548	200
59	15	28	24	10.78	555	400
76	19	13	25	11.28	731	300
52	14	8	26	11.63	485	400
27	9	8	27	11.81	224	400
47	14	30	28	12.60	435	400
68	20	4	29	13.89	654	400
97	29	48	30	14.60	955	500
84	27	4	31	16.02	826	400
79	27	24	32	16.49	774	200
66	23	21	33	16.51	640	100
32	13	42	34	17.13	291	400
57	21	7	35	17.81	552	400

70	26	14	36	18.36	689	100
68	26	22	37	18.41	665	300
91	35	45	38	19.42	917	200
37	16	31	39	19.60	342	500
69	28	21	40	19.75	686	100
99	39	19	41	19.88	1005	400
33	15	47	42	20.31	299	300
75	34	12	43	22.48	766	500
95	42	46	44	22.49	982	500
55	26	27	45	22.52	551	300
46	22	8	46	22.63	449	100
80	36	49	47	22.78	813	100
71	34	31	48	23.62	723	300
57	28	25	49	24.04	579	300
98	46	46	50	24.09	1024	200
86	42	38	51	24.83	902	400
40	22	32	52	25.34	397	200
95	48	15	53	25.65	1005	300
98	52	14	54	26.83	1047	300
23	14	36	55	26.93	209	300
78	44	28	56	28.12	837	500
68	41	17	57	29.80	728	300
86	54	26	58	31.44	952	300
67	44	30	59	32.36	739	200
97	63	2	60	32.36	1098	300
82	57	49	61	34.01	938	100
28	21	36	62	34.59	281	100
87	63	47	63	35.42	1011	300
90	66	39	64	35.42	1054	500
83	61	8	65	35.80	972	300
50	37	36	66	35.82	558	300
25	20	44	67	36.11	256	400
78	59	25	68	36.54	913	200
96	74	40	69	37.32	1148	100
90	70	5	70	37.39	1073	200
87	69	16	71	38.11	1043	500
68	55	9	72	38.47	808	100
52	43	42	73	38.67	609	100
41	35	18	74	39.45	475	100
96	80	7	75	39.46	1186	100
52	44	46	76	39.79	623	200
65	55	22	77	39.94	792	200
89	75	25	78	39.97	1097	400
28	24	47	79	40.50	306	100
77	68	6	80	41.10	967	500
92	82	29	81	41.28	1167	400

82	72	49	82	41.33	1026	400
57	52	47	83	41.95	704	200
83	77	9	84	42.92	1069	200
46	43	22	85	43.07	566	100
94	90	43	86	43.46	1235	400
68	66	40	87	43.95	884	200
71	69	38	88	44.35	928	500
98	97	48	89	44.69	1319	500
43	42	35	90	44.85	537	500
32	32	2	91	44.89	383	400
72	71	41	92	44.96	946	200
52	52	21	93	45.00	673	500
41	42	39	94	45.54	525	200
22	22	18	95	45.65	247	100
53	55	25	96	45.99	698	500
69	74	10	97	46.97	943	100
62	66	19	98	47.06	844	200
77	84	3	99	47.45	1075	200
88	96	40	100	47.51	1242	400
76	84	9	101	47.99	1071	200
35	39	8	102	48.35	460	300
81	91	7	103	48.43	1157	100
76	86	15	104	48.77	1086	400
73	85	24	105	49.44	1059	200
29	34	21	106	50.21	381	200
25	30	26	107	51.01	324	300
47	57	10	108	51.06	669	400
34	42	19	109	51.15	475	100
53	65	8	110	51.54	776	200
65	81	49	111	51.67	980	200
50	64	14	112	52.23	747	300
58	75	14	113	52.93	881	400
52	67	18	114	52.94	783	200
53	70	20	115	53.38	814	200
38	51	32	116	54.40	566	100
36	49	34	117	54.70	541	400
60	84	34	118	55.38	969	100
61	88	38	119	55.88	1007	300
59	86	43	120	56.36	980	300
37	54	44	121	56.69	589	400
46	70	3	122	57.73	769	400
64	99	9	123	57.84	1110	300
36	55	3	124	58.08	593	500
57	89	11	125	58.25	993	100
54	88	20	126	59.33	968	200
47	79	25	127	60.38	855	200

51	89	37	128	61.19	968	500
54	94	40	129	61.30	1022	200
12	19	1	130	61.82	158	300
47	84	42	131	62.15	899	200
13	21	4	132	62.75	187	500
43	82	1	133	63.41	866	100
27	50	28	134	63.89	504	400
42	84	10	135	64.46	876	400
31	61	44	136	65.03	627	200
42	88	21	137	65.51	912	100
23	47	35	138	66.20	463	500
41	92	41	139	67.52	941	200
36	82	31	140	67.71	833	400
38	87	39	141	67.84	890	300
31	71	21	142	67.92	715	300
41	96	2	143	68.06	987	300
35	82	34	144	68.20	829	300
28	63	13	145	68.35	626	100
26	60	10	146	68.43	595	200
40	95	4	147	68.57	968	200
21	47	45	148	68.88	452	200
40	97	13	149	69.30	993	200
19	44	45	150	69.76	422	200
38	96	18	151	70.07	971	100
24	58	16	152	70.13	567	300
28	72	38	153	70.68	716	200
32	88	18	154	71.67	877	500
11	25	25	155	72.82	216	300
24	69	48	156	73.27	670	500
30	92	40	157	73.81	911	200
20	57	32	158	73.92	545	500
22	70	10	159	74.80	682	100
29	99	17	160	75.62	974	100
22	82	45	161	77.23	791	100
18	72	34	162	78.64	690	100
20	82	6	163	78.86	786	500
16	65	22	164	78.93	619	400
15	59	11	165	79.60	554	400
18	76	48	166	79.63	727	100
14	62	19	167	80.16	579	400
8	30	4	168	82.27	260	500
8	29	2	169	82.90	242	200
8	31	8	170	83.33	266	300
8	58	36	171	85.94	537	400
7	41	39	172	86.04	361	400
5	20	35	173	87.38	153	100

8	97	3	174	87.65	926	500
7	88	31	175	88.15	837	500
7	81	11	176	88.27	762	300
5	53	34	177	89.17	484	500
4	56	49	178	90.55	517	400
3	63	23	179	91.08	584	200
3	69	43	180	91.24	648	100
3	83	36	181	91.38	786	300
1	96	30	182	92.08	910	400
0	93	27	183	92.65	887	400
2	61	26	184	92.92	568	100
0	95	34	185	92.98	904	400
3	42	14	186	93.06	374	300
0	81	42	187	93.37	766	400
1	59	23	188	93.65	549	200
2	39	9	189	94.42	350	100
0	51	3	190	95.25	469	500
0	28	35	191	101.36	243	400
15	3	42	192	353.42	104	300
40	0	20	193	353.43	358	100
52	1	18	194	356.30	480	500
81	0	35	195	356.70	765	100
64	1	14	196	356.84	597	200
65	1	10	197	356.99	608	400
77	1	6	198	357.48	726	300
93	2	37	199	358.25	882	500
36	4	3	200	359.63	310	500

Revised 3/21/94.