AD-A243 365

In-MaRC

PHASE I FINAL REPORT

Unique Applications for Artificial Neural Networks

DARPA SBIR 90-115

Contract # DAAH01-91-C-R116

8 Aug 1991

Submitted to

Defense Advanced Research Projects Agency, DSO
3701 North Fairfax Drive
Arlington, Va 22203-1714

From

In-MaRC, Inc
309 East Broadway
Bismarck, ND 58501

Subcontractor

Dr. Kendall E. Nygard
Department of Computer Science and Operations Reseach
North Dakota State University
Fargo, ND 58105-5075

91-17521

Acknowledgments

In-MaRC wishes to acknowledge the significant effort that went into this Phase I project by a number of researchers. Their efforts have contributed substantially to advancement in the field of vehicle routing and scheduling in general, and to real-time dynamic routing and scheduling in particular. The following individuals contributed to the work under this program.

Dr. Kendall E. Nygard
North Dakota State University

Mr. Taesik Kim
North Dakota State University

Mr. Douglas Schesvold
North Dakota State University

Mr. Michael J. Hennebry
North Dakota State University

Mr. Joel D. Melarvie
In-MaRC Inc.

Dr. Sam R. Thangiah
In-MaRC Inc.

A-1

Table of Contents

Unique Applications for Artificial Neural Networks

ABSTRACT

The investigation concerns the application of modular neural networks, working synergistically with genetic search, to provide a powerful means of intelligently controlling heuristic mathematical algorithms for large-scale vehicle routing and scheduling problems. The design lends itself naturally to parallel computing on loosely coupled networks of computers, and to implementation on parallel architectures such as MIMD machines.

Extensive developmental work, coding and computational testing was carried on generic vehicle routing problems. The results are consistently superior to known alternatives, and provide strong motivation to extend the approach into more complex problem domains and military applications. The basic approach was also applied to routing problems with time constraints, a significant complication of considerable practical importance. Results on this problem are also consistently good, and there is potential to further investigate the use of the approach in this domain. Finally, very preliminary results are available for applying the methodology to routing and mission planning for remote autonomous military vehicles, such as Tomahawk cruise missiles or other smart weapons systems. Results are very encouraging, and could be considerably matured during a Phase II project.

In summary, the high performance achieved suggests that multiparadigm approaches that utilize methods from artificial intelligence in conjunction with powerful and proven methods from mathematical combinatorial optimization can build upon the strengths of each constituent, and achieve performance that none of the methods can obtain in isolation.

Keywords: Vehicle routing and scheduling, algorithms, heuristics, neural networks, genetic algorithms, transportation

1.0 Introduction

Efficient routing and scheduling of vehicles is of fundamental importance to many organizations in both the military and the private sector. Primary applications involve the transport of materials and people. However, the mathematical and computational methods also apply to many other problems, including routing message traffic in communication networks, scheduling processes on machines, and to VLSI (Very Large Scale Integration) chip design. Considerable effort has been devoted to developing models and algorithms for vehicle routing and scheduling problems. Material that summarizes progress to date is available [Bodin, Golden, Assad and Ball, 1983; Assad and Golden, 1988]. From a mathematical view, essentially all vehicle routing and scheduling problems belong to the extremely difficult class of NP-complete problems [Fisher and Jaikumar, 1981], meaning that primarily approximation (heuristic) algorithms can be applied to problems large enough to be of much practical interest. In particular, most military applications are amenable primarily to heuristic algorithms, although network optimization, linear programming or other exact methods can and should play a role in some cases.

This investigation involves a multipardigm approach for solving routing and scheduling problems. The paradigms include subsymbolic artificial intelligence methodologies (neural networks and genetic search) as well as combinatorial optimization methods. In essence, the approach starts by applying a modular neural network system to the problem. This system carries out feature extraction and classification, and provides a set of parameters for a mathematical model (heuristic optimizer) as its output. The appropriate mathematical model is run, generating a feasible solution to the problem. In addition, a genetic search is conducted for improved parameters, again with a mathematical model used for evaluation. The mathematical models are time honored procedures for routing and scheduling that have traditionally been driven by paramters chosen in simplistic and static ways. Through intelligent parameter setting, the performance of these models is dramatically improved. The overall approach is outlined as follows:

> **Step 1.** Pre-process the data, representing it so that a neural network system can readily extract salient features and use them to generate parameters for a mathematical model to use in providing a solution to the problem.

> **Step 2.** Apply the neural network system to the problem, producing a set or sets of the parameters needed by the mathematical model.

> **Step 3.** Treating a set of parameters as a populaton member, use paramater sets from the neural network system as a portion of the initial population for the genetic search, creating the remainder at random.

> **Step 4.** Apply the genetic search technique, continuing until a prespecified number of generations has been reached or until a convergence criterion is met. Each population member is evaluated by using its parameter set to drive the the mathematical model.

Important characteristics of the approach are inherent adaptation to problem instance, and dynamic adaptation to the structure of the emerging solution even as the procedure is running. The computational results establish that in at least some problem classes, combining the approaches has a synergistic effect, providing solutions uniformly superior to those obtainable by alternative methods from the literature. The approach readily parallelizes and has considerable potential for real-time dynamic vehicle routing, a problem area of enormous potential for cost-

savings and performance enhancement.

Specific instantiations of the basic approach were developed and applied to several types of problems, including: i) the generic vehicle routing problem (VRP), in which there is a fleet of vehicles with capacities, and stops have demand levels for service, ii) an extension called the vehicle routing problem with time windows (VRPTW), in which the stops have time windows associated with them, and iii) an application to route and trajectory plannng for remote autonomous military aircraft, such as a Navy Tomahawk cruise missile.

To designate specific instantiations of the overall methodology, we use the notation NGO (for Neural network, Genetic algorithm, Optimizer) followed by the notation for the problem addressed. For example, NGO-VRP refers to the hybrid algorithm for specifically solving the VRP.

The computational results are of high significance. For the VRP, a problem that has received considerable attention in the literature, the new NGO-VRP methodology generates better solutions than all of the applicable competing methods on every test problems that we evaluated. Details are provided in Section 2. For the VRPTW, the neural network step will be implemented in Phase II of the research. However, we did employ a genetic search and mathematical model from a random start, and achieved very promising results for this extremely difficult class of problems. Results are described in Section 3. Finally, prototype work was carried out for routing of autonomous military aircraft, with data supplied by the Naval Surface Warfare Center (NSWC) for a Tomahawk cruise missile. Although the missile study is also being carried out under NSWC sponsorship, the basic methodologies of this DARPA investigation are applicable to the problem (and other smart weaponry problems), and clearly illustrate the promise of the new methods in problems of substantial importance in the military as well as the private sector. Section 4 describes the status of the missile routing study. Finally, Section 5 describes future research directions.

We emphasize that the overall approach has little in common with the many reported studies in which subsymbolic AI techniques, such as neural networks, genetic algorithms or simulated annealing are used to directly generate solutions to the traveling salesman problem [Hopfield and Tank, 1986, Liepins and Hilliard, 1989; Kirkpatrick, 1986; Greffenstette, 1988]. Rather, we use neural nets and genetic search as adaptive controllers that intelligently select and steer mathematical heuristics that generate the actual solutions. In this way, the state-of-the-art in combinatorial optimization is brought to bear upon the problems as well as the AI techniques.

2. The NGO-VRP Solver

The NGO-VRP system improves the ability of heuristic procedures to design routing plans, by "intelligently" setting their parameters. The overall system starts with problem data consisting of stop-point locations, depot location, number of vehicles and their capacities. A structural description of each of the functional components is described in this section.

When a given set of model parameters has been set by either the initial neural network system or by the genetic search, we use a "cluster first route second" approach to fitness evaluation for the VRP. There are two fundamental components of this approach. First, the clusters (i.e., assignments of stops to vehicles) must be determined by some method. Second, the stop locations must be efficiently sequenced. The overall objective is to find the order in which the stops are visited so that the total distance traveled by all vehicles is minimized. We experimented with four

mathematical methods of determining the clusters. The first two methods, Fast Assignment Approaches FAA1 and FAA2, are new and relatively fast heuristics. The third method, FGAA, is a modified version of the generalized assignment method of Fisher and Jaikumar (1981). The fourth method, COMBO, is a combination of the other 3 in a multiple sharing scheme. XCHANGE is a simple postprocessor that achieves its power from genetic search. XCHANGE uses a genetic string to directly represent the stop assignments of each route. The effect of the genetic recombinations is to make simple local exchanges to the relative positions of the stops within and among the routes.

2.1 Modular Neural Network System

The modular neural network system that we employ is illustrated in Figure 1. This component operates at a meta-level, selecting the underlying heuristic algorithms and the best parameters to set for the problem at hand.

Artificial Neural Networks draw inspiration from the organization and architecture of human brains as they are presently understood. The paradigm involves the simultaneous examination of numerous hypotheses, and the processing of data in a distributed, concurrent fashion. The interconnections, junction points and weights associated with the interconnections comprise a knowledge base. Neural networks learn by being trained with a large number of instances of the problems that they are intended to address, eventually "learning" the solutions to the training problems. When a new problem is presented to the network, propagation through the network structure is governed by similarity with the problems that were used in training, and an "intelligent" solution is thus obtained through analogy. Because of this activity, artificial neural networks have emerged as a primary artificial intelligence technique for representing a computer-based associative memory [Arbib, 1987].

In our routing system, the basic job of the neural network system is to accept an instance of the routing problem as input, analyze the problem, then provide a collection of parameter settings as output. This is essentially a pattern classification task, a the type of problem solving for which neural networks are well suited [Rumelhart, Hinton and Williams, 1986]. The architecture shown is one that we have developed and used quite successfully for pattern classification in large and complex domains. The mechanism relies on condensation of information through the use of two feature extraction networks that have a "fan-in fan-out" topology (Shown on the left in Figure 1). The idea is to compress the offered data to fundamental features before attempting to classify the problem. On the input side, the raw routing problem data is represented as frequency class vectors for the statistical distributions of geographical locations of the stops, locations of vehicles, and the angles among the locations. These descriptors represent the problem data in a way that is invarient under rotations and scalings. This is done because neural networks are well-known to have difficulty recognizing rotated and/or scaled versions of a pattern. In our experiemntal work conducted thus far, we use 10 frequency classes for each descriptor, yielding an input vector of 30 units. For a 200 stop problem, Figure 2 illustrates the frequency classes for the first descriptor, distance from the depot. The proportions of the stops within each of the bands provides the first 10 units of the input vector. Note that distance clustering relative to the depot is captured by this descriptor, and rotated versions of the same problem would provide precisely the same input vector, which is the effect we seek. The next 10 units are frequency classes for angular dispersion between stops. Thus, the first two descriptors strongly capture clustering of the data in a summary way, known to be a critical factor in vehicle routing parameter setting [Nygard, Kadaba and Juell, 1990]. The final descriptor is relative distance within the bands, providing a fine discrimination capability that may be needed for some problems. Note that even relatively large problems can be readily "pre-processed" into a 30-unit vector of relevant information for the neural networks in this way. This 30-unit problem data vector is compressed into a small vector at the center of the feature extraction neural network.

This is accomplished by setting the output layer of the neural network to values that are identical to the input layer, essentially letting the network self-organize the information during training. A similar network operates on the performance (output) side of the system, seeking a condensed representation of appropriate parameter settings.

Identifying a successful representation of a parameter set as a bit string that can be generated by the neural networks and serve as an artificial chromosome is a key issue in the approach. The control parameters for the VRP are seed-point coordinate locations [12]. We model each vehicle tour with a location called a seed, with the vehicle conceptually traveling from the depot to the seed and back. The seed-points represent a nominal direction and distance from the depot that the vehicle travels. We use binary bit strings to encode the seed-points as shown in Figure 3. The number of seed points is equal to the number of vehicles available. The example string shown below represents three seed-points, each with an x and y coordinate, shown in both binary and decimal. The control parameters S_{x1}, S_{y1}, S_{x2}, ... S_{xN}, S_{yN} represent the N seed-points. The parameters are constrained to an interval of the form $a_i \le S_N \le b_i$ where a_i is 0 and b_i is 1023. Each seed-point is identified by an x-coordinate and a y-coordinate expressed in a 10-bit binary string, allowing representation of the integers 0 to 1023 inclusive. The delivery locations for the actual problem are scaled to a 1024 X 1024 grid with only integer positions possible. Although the GA operates at the fine-grained resolution of the 1024 X 1024 grid, the neural network system operates at the coarser-grained resolution of the 5 X 5 grid illustrated in the Figure. The resultant 25 grid locations are represented as a vector of zeroes and ones, indicating the presence or absence of a seed point. Figure 3 illustrates the encoding of an example.

Finally, as shown on the far right of Figure 1, a generalization network operates with input and output layers extracted from the centers of the two feature extraction networks. With this architecture, the generalization network uses the summarized essence" of the problem on the input side, and the summarized "essence" of the mathematical solution approach on the output side.

During actual operation with the trained network system on a new problem, the generalization net must provide detailed recommendations for parameter settings. This is accomplished by feeding the summarized output settings into a reconstruction network that is comprised of precisely the bottom half of the performance feature extraction network. The reconstruction network is illustrated with dotted lines on the far right in Figure 1. This is significant departure from the standard approach in which new problems are processed with precisely the same generalization net used in training. We have experimented with many single network configurations of layers, connectedness, and units per layer for this problem. In classification accuracy in identifying the best heuristic, we have clearly established that the modular system is vastly superior to single netowrk topologies [Nygard and Kadaba, 1990]. The approach parallels the following steps that a human expert might use in reasoning about the solution approach to a new problem:

Step 1. Represent and view the data in such a way that recognizable and meaningful patterns are identifiable (analogous to the frequency classes chosen).

Step 2. Identify the essential global features of the problem (analogous to feature extraction on the input side).

Step 3. Use the global features identified in step 2 to identify key global features of promising approaches to the problem. (analogous to feature extraction on the output side).

Step 4. Pursue the selected solution approach in detail, setting values for the parameters that govern the solution process.

Step 4 is initialized by the modular neural networks, but extends naturally into a genetic search,

described in the next section.

All training was conducted on an IBM class 3090 supercomputer. The training set consisted of sets of sample problems that the genetic search was applied to with a random start and run until no further improvement was possible. After training, the weights were downloaded to a SUN workstation computer for test runs on previously unseen problems. Overall, the neural network system expedited convergence of the method by about a factor of 3.
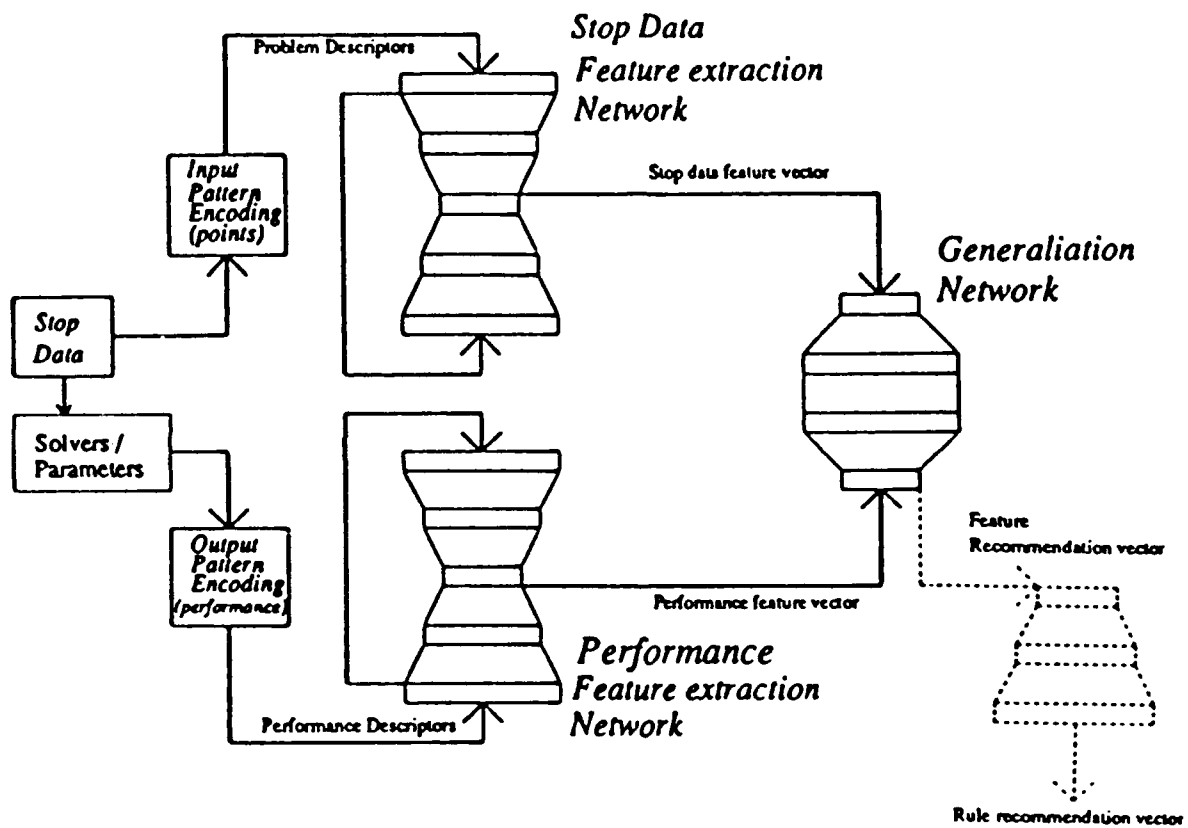
Figure 1. Overall structure of the neural network system. Feature extraction networks are used for both the problem description data and the algorithm performance data for sets of parameters. The generalization network operates with the condensed information at both the input and output layers.

Figure 2. Frequency classes used to encode distance from the central depot. For each problem, the proportions of the stops within each band are input descriptors for the neural network system.

Encoding: 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0

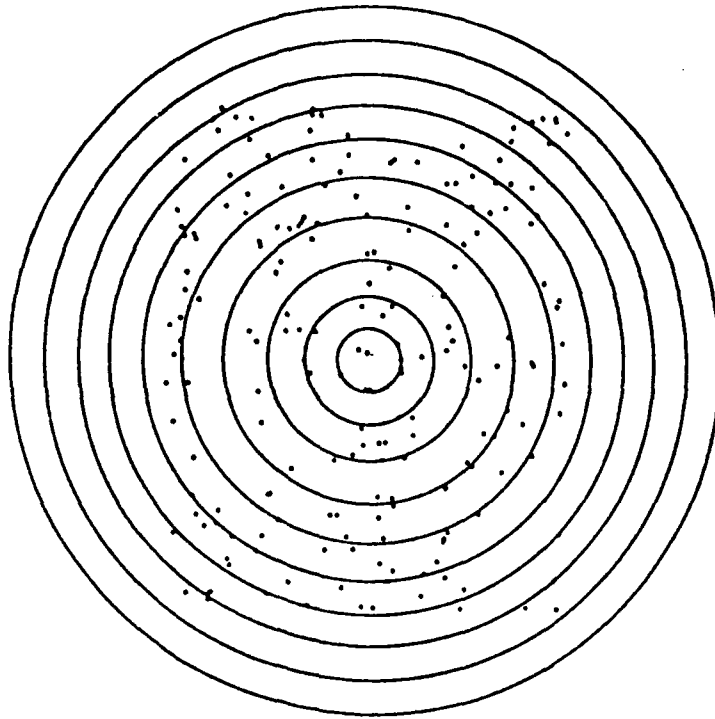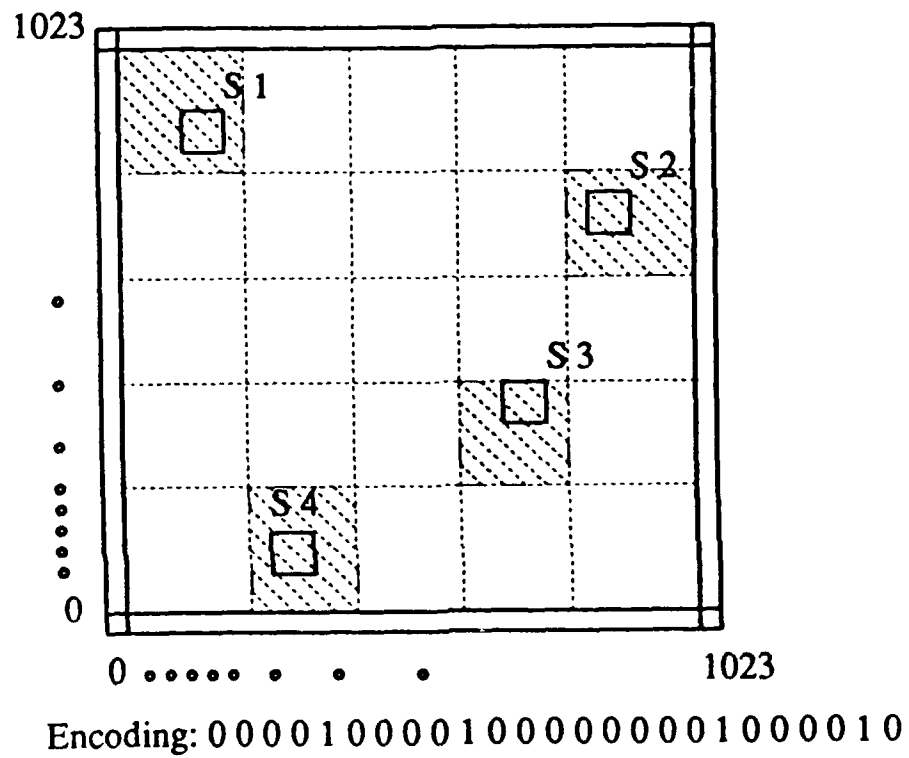**Figure 3.** Ecoding of the parameters. The neural network is coarse grained, fuictioning on the 5 X 5 grid. The genetic search operates at the 1023 X 1023 resolution.

2.2 Genetic Search

Genetic search algorithms are heuristic solvers that proceed in a manner inspired by biological genetics. The basic concepts of genetic algorithms were primarily developed by Holland [75]. The book of [Goldberg, 1989] is a good survey of adaptive genetic search. A pseudcode representation of a genetic algorithm is shown in Figure 4.

```
Generation <- 0;
Initialize M(Generation);
Evaluate M(Generation);
LOOP (until termination condition)
        Generation <- Generation + 1;
        Select M(Generation) from M(Generation-1);
        Crossover M(Generation);
        Mutate M(Generation);
        Evaluate M(Generation);
LOOP END
```

Figure 4. Pseudocode for a genetic algorithm

In this study, a modification of the GENESIS code developed by John Grefenstette of the Naval Research Laboratories is used for all of the computations [Grefenstette, 87]. GENESIS was chosen because source code written in C is available and it is well-regarded by genetic algorithm researchers. The genetic algorithm (GA) is an iterative procedure that maintains a population of P candidate solutions over many simulated generations. The population members are bit strings that serve as artificial chromosomes. The chromosomes are fixed length strings with a finite number of binary values (or alleles) at each position (or locus). The initial population M(0) consists of P members whc ᵥ ᴊits are chosen by the neural network and at random. Each chromosome is evaluated by a mathematical model, resulting in a fitness or cost value. The fitness value determines the relative ability of a chromosome to survive and produce offspring in the next generation.

The selection operation, Select M(Generation), of a chromosome, C_i, where i = 1 to P, into the next generation is dependent on its fitness value f_i relative to the fitness value of other chromosomes in the population. At (Generation - 1) the relative fitness of the chromosome, C_i, is calculated as follows: $Relative \ Fitness \ of \ C_i = \frac{f_i}{\sum f_i}$ In carrying out the selection process, we treat the relative fitness value as probabilities, and calculate the expected number of copies of each chromosome, C_i, chosen for producing offspring to survive into the next generation as follows:

*Expected number of copies of C_i = Relative Fitness of C_i * P*

For example, if the expected number of copies of C_i is 0.2 and P is 100, then 20 identical copies of C_i will survive for possible reprodution into the next generation. The expected values are processed in descending order until the population size is reached. After the selection procedure, population M(Generation + 1) contains exact duplicates of the selected structures from population M(Generation). 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, called the crossover ratio (*C_Rate*), undergoes a two point crossover operation and produces offspring for the next generation. Crossover exchanges alleles among adjacent pairs of the first (*C_Rate * Population Size*) chromosomes in the new

-9-

population. The remaining proportion of the selected chromosomes are carried over without change to avoid massive disruption of the population. A two point crossover proceeds in two steps. First, two adjacent parent chromosomes are chosen. Second the two chromosomes exchange chromosome bits as follows: two integer bit locations, say p and q, are chosen randomly, where p and q lie within the range of the chromosome length and p is greater than q. Two new offspring are formed from the parent chromosomes by exchanging all bits that lie between p and q

Selection and crossover effectively search the space exploring and exploiting information present in population chromosomes by selecting and recombining primarily those offspring that have high fitness values. These two processes eventually produce a population of chromosomes with high performance characteristics. The Mutatation operator, M(Generation), is a secondary operator that prevents premature loss of important information by randomly mutating bits within a chromosome. Mutation proceeds by selecting a proportion of the chromosomes, defined by the mutation ratio (*M_Rate*), and converting the chromosome bits to their complementary value. Approximately (*M_Rate* * *Population Size* * *Chromosome Length*) mutations occur per generation. GA's are adaptive in the sense that candidate offspring chromosomes generated at each generation reflect and exploit information obtained by chromosomes of earlier generations. The adaptiveness is achieved by exploiting similarities present in the coding of the chromosomes. The termination criteria of the genetic algorithm is absence of improvement within a prespecified number of evaluations or when a maximum number of generations is reached.

Methods of improving the performance and efficiency of GA's are of significant importance. The primary parameters of a standard GA are population size, crossover and mutation rates, and number of crossover points. These parameters have a significant impact on performance [17, 7, 11]. Adaptive selection methods [1] and reproductive evaluation techniques have also been shown to speed up GA searches. Genetic searches are generally compute intensive procedures that require the evaluation of many candidate solutions to a given problem. In the application area we study (routing and scheduling), the genetic algorithm sets parameters for a mathematical heuristic. To reduce the computational overhead of this approach, we developed a mechanism for improving the performance of the genetic search. We employ a method of using multiple sharing evaluation functions, permitting the parallel investigation of multiple peaks in the search space.

2.2.1 Fitness Evaluation

We use several "cluster first route second" heuristic techniques to evaluate fitness in the GA. The first phase, clustering, determines the sets of stops to be served by each vehicle. The second phase, routing, sequences the stops assigned to each vehicle.

To handle the first phase, we developed and experimented with four methods of determining the clusters. The first two methods, Fast Assignment Approaches FAA1 and FAA2, are new algorithms that are relatively fast and well suited for the genetic search. The third method, FGAA, is a modified version of the generalized assignment method developed by Fisher and Jaikumar [4]. COMBO, the fourth method, employs the other three in combination, with each driven with the same set of parameters.

In the second phase, a Traveling Salesman Problem (TSP) solver is used for sequencing. In the TSP, a tour begins at a home location, visits each stop on a list exactly once, then returns to the location of origin. The objective is to find the order in which the stops are visited so that the total distance traveled is as small as possible. Tour construction algorithms are a prominent and successful class of mathematical heuristic procedures for quickly solving large-scale instances of the TSP These methods construct tours incrementally, starting with an initial subtour, then expanding it by repeatedly applying rules that select unvisited stops and insert them into the tour. In this work, we utilize only the CCAO tour construction algorithm shown by Golden and Stewart to consistently achieve high performance with reasonable CPU times [8].

-10-

In summary, evaluation of a population member proceeds as shown in Figure 5.

Step 1:   Input the seed point coordinates represented by
            the chromosome of the population member

Step 2:   Clustering Phase. Use FAA1,FAA2 FGAA or COMBO
            to assign stops to vehicles

Stage 3:   Sequencing phase. Use the TSP heuristic to order
            the stop points in each cluster

Step 4:   Calculate total tour length for each cluster, sum over all clusters,
            and return the total as the fitness value

Figure 5. Evaluation of a population member.

All the clustering methods have merit, in the sense that each induces high performance on at least some example problems. Each of the clustering methods is described below in turn.

Clustering Method FAA1: In this method, one seed-point is active at a time. The nearest stop is assigned to the active seed-point, if doing so does not violate the corresponding capacity constraint. For each stop assigned, a weighted distance factor is added to the active seed-point. The seed-point with the minimum weighted distance is made active for the next assignment. This process continues until all the stop points are assigned to some seed-point.

Clustering Method FAA2: In the second method, all the seed points are eligible to receive the next stop point assignment. in At each step, the stop point with the minimum distance to any of the seed point is selected and assigned to that seed point. The process continues iteratively until all the clusters are formed.

Clustering Method FGAA: In FGAA, a generalized assignment optimization problem is solved to assign the stops to the seed-points. The basic idea is due to Fisher and Jaikumar [4], and is widely held to be the most consistent high performance approach to the generic vehicle routing problem. Using genetic search as an intelligent shell to find high performance seed point values significantly increases the ability of the generalized assignment method to calculate high performance clusters. Given a set of seed points, the following generalized assignment problem is solved to assign stops to vehicles.

$$minimize \sum_{k \in K} \sum_{j \in J} c_{kj} x_{kj}$$

subject to:

(1) $\sum_{j \in J} r_{kj} x_{kj} \le b_k$ for *all* $k \varepsilon K$

(2) $\sum_{k \in K} x_{kj} = 1$ for *all* $j \varepsilon J$

$x_{kj} = 0 \; or \; 1$ for *all* $k \varepsilon K, j \varepsilon J.$

The value of the decision variable, x_{kj}, is interpreted as follows:

$$x_{kj} = \begin{cases} 1 \text{ if } stop \ j \ is \ assigned \ to \ vehicle \ k \\ 0 \ otherwise \end{cases}$$

Constraint set (2) forces each stop to be assigned to exactly one vehicle. Constraint set (1) limits the assignments by vehicle capacity.

With this model, the cost parameters are defined in terms of distances D as follows:

$$c_{kj} = D(depot, stop \ j) + D(stop \ j, seed \ k) - D(seed \ k, depot).$$

This definition of c_{kj} provides a linear model of the contributions of individual stops to the travelling salesman tours. The clusters produced in Clustering Step are fundamentally dependent on the locations of the seed points, and provided much of the inspiration to intelligently search for high performance seed points.

COMBO Clustering Method: Many optimization problems require the investigation of multiple local optimas. Here a concept inspired by sharing functions (Goldberg 1987) is used to investigate the formation of stable subpopulations of different strings in the GA, thereby permitting the parallel search of many peaks. This method uses the string recommended by the GA on all three methods (FAA1,FAA2,FGAA) and the function with the best performance value is selected to return the fitness value for that particular string. The three methods (FAA1,FAA2,FGAA) all use the same string, and due to the competition between widely disparate points in the search space, help maintain a diverse population which searches many peaks in parallel. This multimodal optimization method also helps in avoiding premature convergence due to local optimas.

2.2.2 Convergence characteristics

Figure 6 and Table 1 collectively illustrate the mechanics of a crossover genetic operator. Under 2-point genetic crossover with random crossover point selection, suppose that C_{11} and C_{12} are the crossover points in Parent 1 and C_{21} and C_{22} are the crossover points in Parent 2. When the genetic material (bit strings) between the crossover points exchanged, two offspring strings are generated. The crossover operator results in two candidate seed-point locations, with seed-point 3 not being effected by the crossover operation, but seed-point 1 and 2 moved to a new location. These new locations of the seed-points result in altered clusters, which in turn results in a altered fitness measure (total distance). After sequencing the stop locations in each of the candidate clusters with the TSP solver, the results are returned to the GA.

As generations evolve, seed-points tend to be concentrated in tight geographical areas due to the survival of the fittest mechanism of the GA. This is illustrated in Figure 7 for a typical example problem. In part a of the Figure, seed point locations for the last 50 of 1000 trials are plotted with distinct symbols for each seed. In part b, all 1000 trials are plotted. A trial is a single execution of the evaluation function on a candidate control parameter. From the plots it readily seen that the seed locations settle into high performance geographical areas.

The three performance curves on the graph shown in Figure 8 illustrates the the convergence of the search process. The performance measure is total distance traveled by the fleet, so small values are desirable. The top curve indicates the worst performance of the evaluation function as function of generations. The bottom curve indicates the best performance in each generation. The middle curve is the plot of the average performance of the evaluation function. The decreasing trend in the curves illustrates the survival of the fittest candidates in the population, and indicates that the GA is doing much better than a random search in the control parameter search space.

Table 2 presents empirical work that illustrates the parallel nature of the search and characteristics of the multiple sharing evaluation functions used in the COMBO clustering method. The illustration is for a typical four vehicle problem. A row shown in the Table corresponds to an evaluation function (FAA1, FAA2, or FGAA) evaluating a seed-point parameter which produces a performance value better than the best found up to that point in time. Reading from the top, the FAA1 method produces the best solution initially (row 1). The next seven improving solutions are found by FAA2 (rows 2 through 8). The seed-points that produce these solution are the result of searches centered around a few relatively high performance locations. The ninth improvement is found at generation 14 by FAA1. The coordinate value reveal that this seed-point location is very close to the one FAA2 was using to improve the solution performance, as shown in example 9. The FGAA method, which had not generated a best solution among the first 10, produces an improved solution in generation 17 using seed-points very similar to a pattern first identified in generation 2. Also note that the seed-points in example 11 are in a completely different area of the search space. This illustrates the parallel search of the multimodal response surface occurring in the search. Thus, each of the evaluation functions is able to take advantage of progress made by the other functions at any generation.

| The Mechanics of the Crossover Operator						
String	$S_{1,1}$	$S_{2,1}$	$S_{1,2}$	$S_{2,2}$	$S_{1,3}$	$S_{2,3}$
Seed-Points	100	100	900	100	900	900
Genetic Encoding	0001100100	0001100100	1110000100	0001100100	1110000100	1110000100
Parent 1	0001100100	0001100100	c_{11}1110000100	0001 c_{21} 00100	1110000100	1110000100
Parent 2	0001100100	c_{21} 0001100100	1110 c_{22} 00100	0001100100	1110000100	1110000100
Offspring 1	0001100100	0001100100	0001100100	1110000100	1110000100	1110000100
Decode 1	100	100	100	900	900	900
Offspring 2	0001100100	1110000100	0001100100	0001100100	1110000100	1110000100
Decode 2	100	900	100	100	900	900
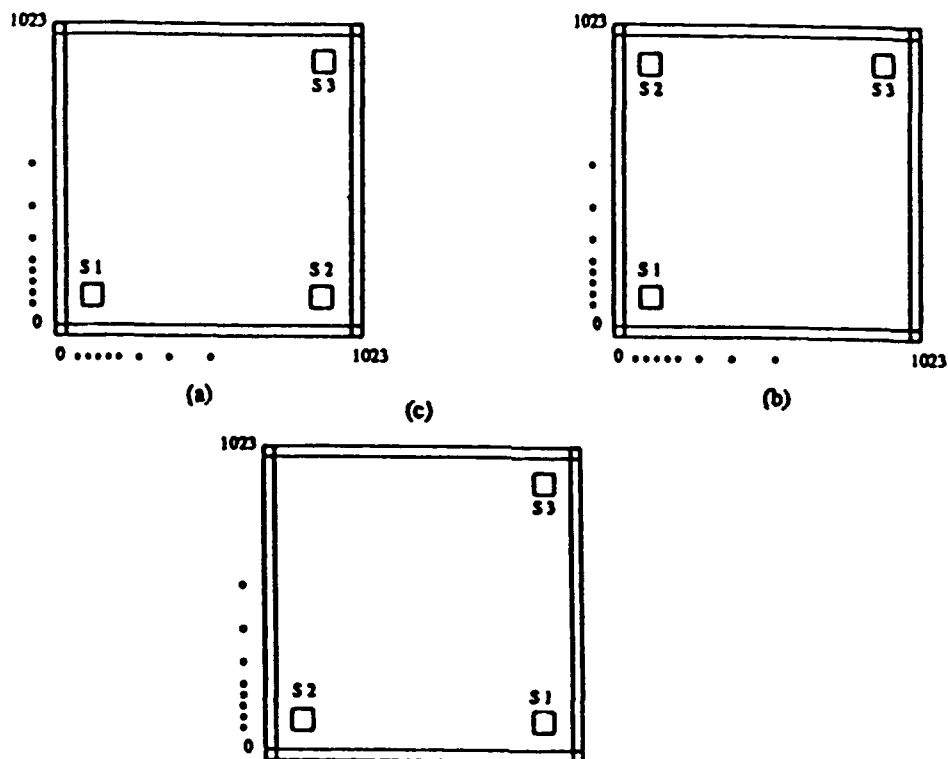
Figure 6. Seed points resulting from a crossover operation wtih data from Table 1. Part (a) shows 3 seed-point locations for the Parents. Parts (b) and (c) show the locations of the two offspring.

Table 2. Parallel nature of the adaptive search. Each of the three methods is able to exploit promising seed-points discovered by the other methods.

| Ex	S_{x1}	S_{y1}	S_{x2}	S_{y2}	S_{x3}	S_{y3}	S_{x4}	S_{y4}	Perf	Method	Generation
1	572	61	959	623	742	43	125	463	12373	FAA1	1
2	812	824	316	528	981	405	181	816	12082	FAA2	2
3	759	851	85	371	49	136	968	279	11880	FAA2	2
4	580	928	105	396	963	818	82	275	11685	FAA2	2
5	466	716	805	75	114	769	494	873	11518	FAA2	2
6	279	488	865	65	51	747	944	660	11132	FAA2	2
7	232	791	865	79	901	672	197	720	10958	FAA2	2
8	757	329	110	833	55	136	968	663	10761	FAA2	7
9	714	182	90	841	52	141	976	652	10732	FAA1	14
10	714	342	105	833	55	128	1006	648	10606	FAA2	16
11	232	151	873	185	53	795	958	464	10490	FGAA	17
12	232	150	873	185	54	868	958	464	10450	FGAA	30
13	773	181	150	185	55	868	945	431	10394	FGAA	32
14	688	197	118	185	55	868	977	524	10373	FGAA	35
15	693	169	83	838	72	151	943	908	10299	FAA2	38

(a)

(b)
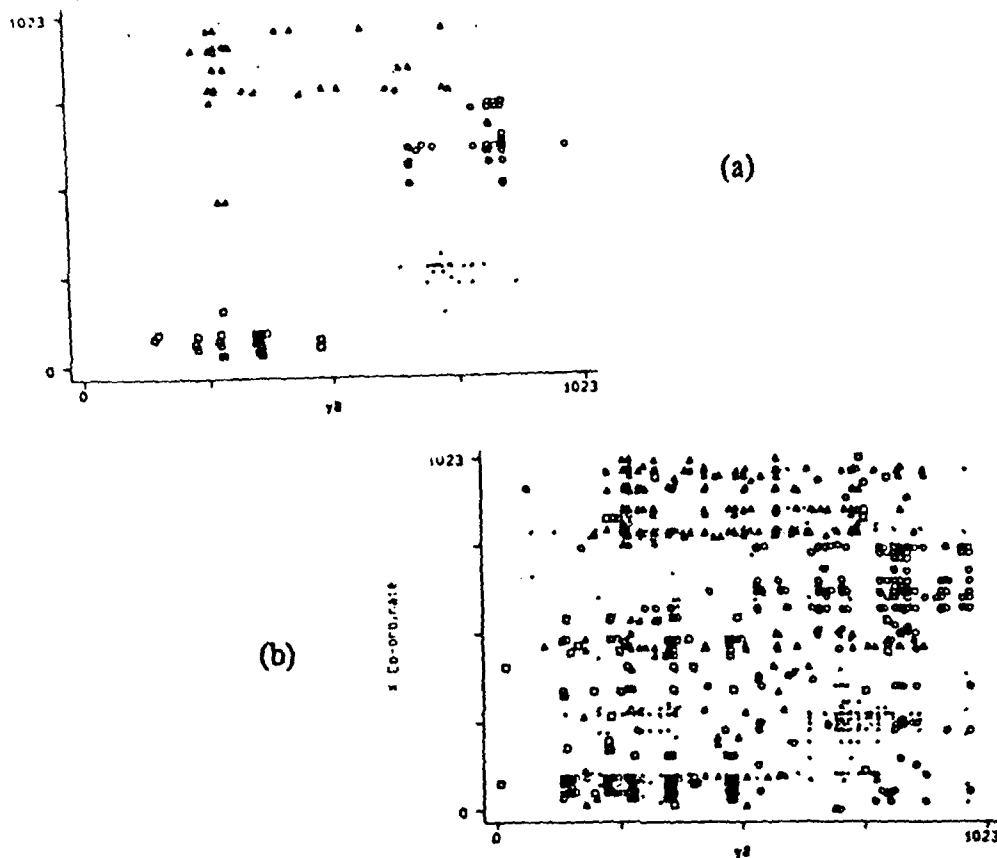
Figure 7. Convergence of sets of 4 seed-points to locations that correspond to high performance by the mathematical models. Part (a) shows the last 50 trials and part (b) shows 1000 trials.
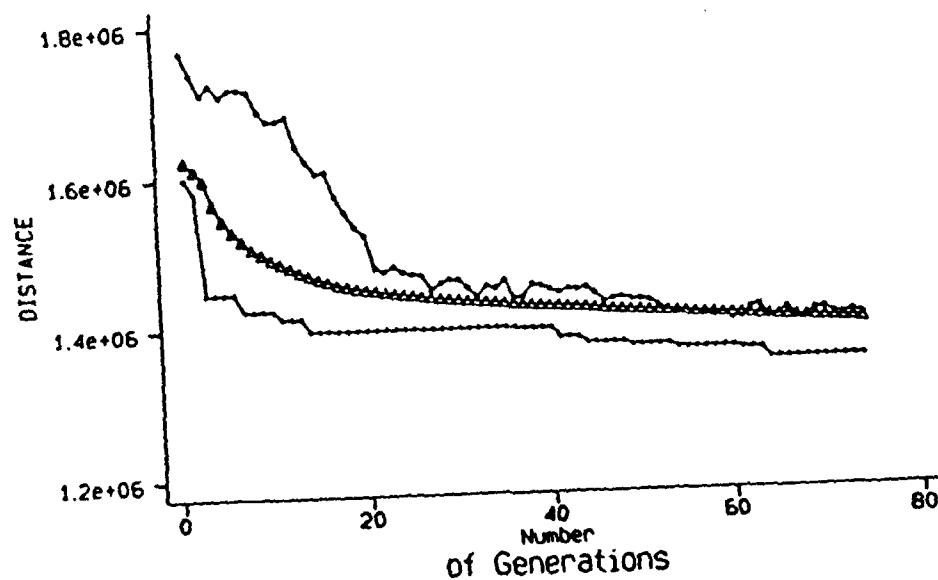

Figure 8. Typical performance of the genetic algorithm on a single test problem. As generations are simulated, both the best solution (bottom curve) and the worst soluton (top curve) show steady improvement. The smooth middle curve is the average. The decreasing profiles show that the regions of the search space being exploited is being reduced in a non-random fashion.

-15-

2.2.3. A Genetic Search Post-Processor

In addition to the 4 clustering methods and the CCAO travelling salesman problem solver, we also experimented with a local improvement procedure designed to be applied as a post-processor. Called XCHANGE, the method is quite simple, but achieves significant results through genetic search. XCHANGE uses a chromosome that directly represents the assignments of the stops to vehicles. Crossover operations and a very small mutation rate make local changes to the stop/vehicle assignments. Each local change is accepted only if the modification would not violate the capacity constraints of the vehicles involved. The procedure continues for a preset number of trials.

Figure 9 illustrates the artificial chromosome used by XCHANGE. Each stop-point is mapped to a cluster number which is represented in binary. In this way, a 2-bit binary string can handle up to four vehicles indexed by 0 through 3 inclusive. Similarly, a 3-bit string could handle up to 8 vehicles. Standard 2-point crossover is employed. The crossover action leaves some vehicle clusters unaltered and exchanges or offloads stops in others.

2.2.4 Experimental Results

Five variations of the overall methodology of using genetic search for parameter setting were empirically evaluated on a large number of test problems. The variations correspond to each of the three rudimentary clustering methods (FAA1, FAA2, and FGAA), the COMBO clustering method, and the COMBO method followed by the genetic postprocessor, called XCHANGE in the ensuing. In addition, each test problem was was also solved with each of 5 algorithms that represent the state-of-the-art from the literature.

The five algorithms from the literature are:

- Clarke-Wright, a venerable heuristic algorithm capable of producing fast solutions to large scale problems with multiple vehicles and an objective of minimizing total distance [3].

- FISHER1, the original generalized assignment heuristic of Fisher and Jaikumar [4] that is well known to produce high-quality solutions to small and medium size problems.

- FISHER2, a heuristic similar to FISHER1 that uses a seed setting strategy that employs a look-ahead feature [13].

- NYGARD-WALKER, a heuristic that uses spacefilling curves and Lagrangian relaxation to obtain solutions to very large-scale multiple vehicle problems [15].

- LFMO, a vehicle routing solver based on Spanning Trees and Branch Exchanges [14].

A random problem generator was used to generate sets of fully dense problems with 100, 200 and 1000 stop points generated in a square, 1023 miles on a side. A vehicle utilization factor of 95 % was used in all problems. The performance measure for a candidate solution is the total distance traveled by all the vehicles. A mutation rate of .001 was used in all cases, meaning that each bit in the representation of each population member is changed (from 0 to 1 or vice versa) with a probability of .001. All experiments were set for 1000 trails (generated feasible solutions). The CCAO selection/insertion heuristic of Golden and Stewart [8] was used to calculate the traveling

salesman tours. The best solution found in the 1000 trials was retained. A great deal of experimental work was carried out on a network of SUN 3 workstation computers. Here we discuss a set of 25 test problems, all fully dense with 200 stop points, and 4 vehicles with a utilization factor of 95 %. Each of the 5 genetic-based methods was run for 1000 trials. All of the GA parameters (crossover rate, mutation rate, population size and string length) were kept constant through out the experiments. Figure 9 illustrates the advantage that the method that uses multiple sharing evaluation functions (COMBO method) has over the methods with individual evaluation functions. By definition, the COMBO strictly dominates each of the alternative methods. Note, however, that the three individual methods do not improve the search after about 1000 trials, but the performance curve of the COMBO method continues to drop through subsequent trials. This result is consistent over all 25 problems tested.

Table 3 provides the final results for all 10 available methods on all 25 problems. The values shown in the table are the total miles traveled using the four vehicles. The XCHANGE method of the table applies the post-processor module to the best assignment achieved from the COMBO method as a initial starting point for the genetic search. The results show that there is a strong tendency for the methods that employ the genetic search to outperform the 5 methods from the literature. In addition, the COMBO method consistently dominates each genetic search controlled heuristic in isolation, and XCHANGE significantly improves the result even more. The percentage decrease in total distance travelled over the range of the methods tested is commonly in the 5% to 10% range, a result quite remarkable in light of the long history of some of the methods represented.

The Friedman test was used to statistically test whether the different heuristic algorithms have equal mean performance measures. This test is a nonparametric counterpart of the parametric two-way analysis of variance (ANOVA) test, and applies if the hypothesis that the data comes from a normal density is rejected. The test can be applied to 3 or more algorithms at once and may be used if only rank data are available. The data are set in a randomized block design with n problems each containing k algorithms. The measurements are ranked in each problem over the algorithm. After doing this for each problem, the ranks were summed for each algorithm. In the case of ties, average ranks were used. The null hypothesis is that all the algorithms (three or more) have equal mean costs and the alternative hypothesis is that all the algorithms (three or more) do not have equal mean costs. Using the multiple comparison test, the conclusion is the XCHANGE algorithm may be regarded as superior to the COMBO algorithm, COMBO is superior to FAA2, and FAA2 is superior to the FGAA algorithm.

To illustrate the low memory requirements of the genetic search methods large routing problems, test problems were generated with 1000 stop points and 4 vehicles were used with a utilization factor of 90%. The genetic algorithm requires little memory and achieves good solutions within seconds, but can require substantial computer time (perhaps several hours) to achieve top quality solutions. However, the genetic algorithm lends itself naturally to asynchronous parallelization on a network of workstation computers. For 1000 stop problems, the only method from the literature that could run with the 8 megabytes of memory is the Clarke-Wright method, and it requires about 6 megabytes, even though extremely efficient data structures were used [13]. The COMBO method requires about 1 megabyte of memory for such problems. In many of the experiments, we used a parallelizing scheme in which as many as 6 asynchronous workstations were used to solve individual problems. Essentially, the fitness function evaluations were allocated to separate processors as new parameters became available. The NGO-VRP methodology outperformed the Clarke- Wright by nearly 10% in reduced mileage over the set of problems tested. Thus, the new methods offers geneuinely large scale capablilities that the best of the competing methods cannot approach. The low memory requirements of the genetic search methods sets the stage for networks of microcomputers or MIMD computers to be brought to bear on very large routing and scheduling problems.

Table 3. Illustration of the performance measures of the various models used to benchmark NGO-VRP. Values shown are the total miles traveled using four vehicles. Overall, previously existing methods are consistently outperformed.

| 200 Node - 4 Vehicle - 95% Utilization									
PROBLEMS	CLRK	FISHER1	FISHER2	NYWK	LFMO	FAA1	FAA2	FGAA	COMBO	XCHANGE
p1.1	11501	10887	11123	11025	11340	10755	10823	10656	10670	10600
p1.2	11522	11251	10943	11418	11003	10532	10273	10230	10215	10163
p1.3	11345	10821	10967	11150	10834	10484	10611	10732	10252	10248
p1.4	11174	10976	10948	12046	10636	10423	10182	10461	10212	10129
p1.5	10950	11307	10863	10892	10697	10726	10233	10593	10145	10021
p1.6	11807	11306	12357	10756	11285	11247	10931	11121	10629	10629
p1.7	10936	10792	10796	10643	10614	10447	10290	10731	10299	10249
p1.8	10596	10716	10758	10872	10356	10098	10108	10178	9994	9905
p1.9	11542	10616	11234	11525	10825	10267	10301	10231	9904	9904
p1.10	10003	10829	10879	10056	10532	9584	9472	9950	9472	9472
p1.11	11023	10621	11014	10845	10509	10350	10176	10489	10130	10125
p1.12	11501	10841	10827	12053	10602	10513	10440	10670	10354	10313
p1.13	11883	11211	11253	11028	11127	10697	10518	10916	10498	10498
p1.14	11254	10961	11113	11763	11425	10501	10558	11047	10521	10519
p1.15	11187	10441	10567	10982	10950	10572	10175	10860	10100	10100
p1.16	11058	11363	11363	11187	11370	10810	10596	10902	10884	10790
p1.17	11665	11250	11250	10968	10934	10520	10375	10870	10261	10200
p1.18	11114	10189	10190	10481	10778	9986	9900	9810	9549	9475
p1.19	11448	10643	11002	10360	10649	10597	10129	10472	10207	10135
p1.20	12538	10885	11063	11846	11686	10715	10660	10784	10755	10755
p1.21	11883	10938	11124	11060	10468	10408	10539	10225	10225	10225
p1.22	10963	10586	10494	10174	10174	9900	9713	9713	9765	9765
p1.23	11212	10638	10776	11237	10632	10742	10599	10619	10558	10558
p1.24	11778	10967	11216	11412	11481	10952	10818	10875	10491	10491
p1.25	11570	10636	10595	11186	10766	10535	10440	10585	10358	10358
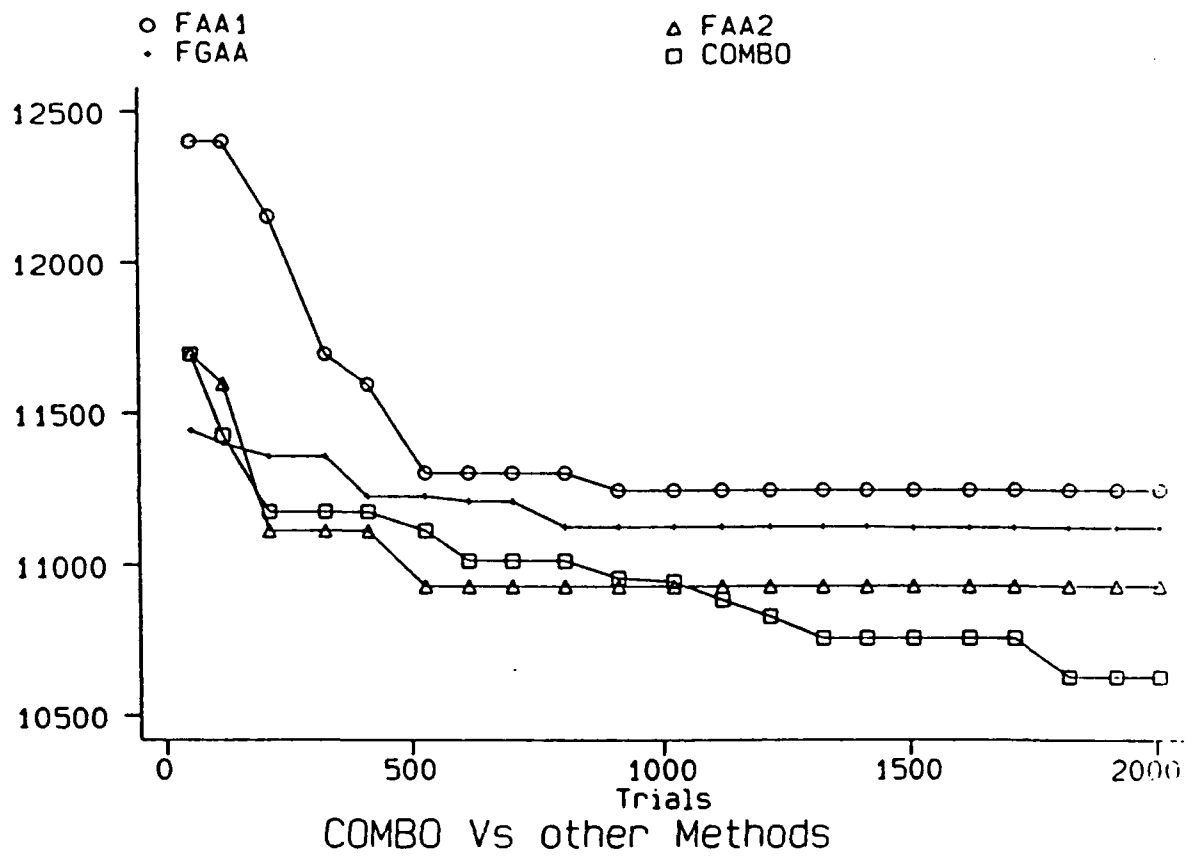
Figure 9. Convergence rates of the methods that were tested. The COMBO method continues to show improvement after the alternatives have stabalized.

3. The NGO-VRPTW Procedure

The VRPTW involves routing a fleet of vehicles, of limited capacity and travel time, from a central depot to a set of geographically dispersed customers with known demands within specified time windows. The time windows are two-sided meaning a customer must be serviced at or after its earliest time and before its latest time. If a vehicle reaches a customer before the earliest time it results in idle or waiting time. A vehicle that reaches a customer after the latest time is tardy. A service time is also associated in servicing each customer. The route cost of a vehicle is the total of the traveling time (proportional to the distance), waiting time and service time taken to visit a set of customers.

The VRPTW arises in a wide array of practical decision making problems. Application includes retail distribution, school bus routing, mail and newspaper delivery, municipal waste collection, fuel oil delivery, dial-a-ride service, airline and railway fleet routing and scheduling. Efficient routing and scheduling of vehicles can save government and industry many millions of dollars a year. Solomon[84] provides an excellent survey on vehicle routing with time windows.

Savelsbergh[85] has shown that finding a feasible solution for a VRPTW using a fixed fleet size is NP-hard. Due to the intrinsic difficulty of the problem, search methods based upon heuristics are most promising for solving practical size problems [Baker, 86] [Solomon, 87][Thompson, 88].

The NGO-VRPTW is an instantiation of the overall approach as it applies to the VRPTW. The genetic search potion and the optimization portion have been developed, coded and empirically evaluated. The genetic search begins with a random start. When the neural network is employed in initialization, we expect that convergence of the process will be on the order of 3 times as fast as the present procedure.

The genetic search proceeds globally, assigning stops to vehicles by a process we call genetic sectoring (GENSECT). Once stops are assigned to vehicles a local route optimization module (LOCOPT) is invoked. The GENSECT module adaptively searches for sector rays that partition the geographical area into sectors or clusters served by each vehicle. The GENSECT solution ensures that each vehicle route begins and ends at the depot and that every customer is serviced by one vehicle. The LOCOPT module operates by removing tardy customers and reducing infeasibilities in capacity and travel time of vehicles. The synergy between a global adaptive heuristic search combined with local optimization gives solutions superior to those of competing heuristic algorithms. On a standard set of 56 VRPTW problems obtained from the literature, NGO-VRPTW did better than the alternate methods on 41 of them, with an average reduction of 3.9% in fleet size and 4.4% in distance traveled. NGO-VRPTW required an average of 127 cpu seconds to solve a problem on the SOLBOURNE 5/802 computer. In addition to performance and efficiency, the NGO-VRPTW system can be used to evaluate the tradeoff between fleet size, vehicle capacity, distance and route time by parametrically varying the values.

3.1. The GENSECT and LOCOPT Modules

NGO-VRPTW consists of two distinct modules, GENSECT, a global vehicle routing module and LOCOPT, a local route optimization module. The GENSECT module assigns the customers into vehicle sectors or clusters. The clustering ideas in GENSECT are inspired by the work of Fisher and Jaikumar[81]. A genetic algorithm, GENESIS [Grefenstette, 87], is used to find a set of sector rays to assign customers to clusters. Once the clusters are formed, the vehicles are routed to serve the customers within a cluster using a cost function that minimizes the total route cost. The vehicle routes from the GENSECT module could consist of tardy customers and infeasibilities in capacity and travel time of vehicles. The LOCOPT module uses local

optimization procedures to move or exchange customers between clusters to remove the tardy customers and vehicle infeasibilities. The iteration between GENSECT and LOCOPT can be performed any number of times. We have found that two iterations are sufficient to attain relatively good solutions.

3.1.1 The GENSECT Module

The GENSECT module initially divides the customers into sectors or clusters using a sweep algorithm [Gillet and Miller, 74] with sector rays as the points of division. Once the clusters are formed, the vehicles are routed to serve the customers while minimizing the route cost calculated by the route cost function.

The cost function gives high priority to reducing tardy customers and penalizes vehicles not within their capacity and travel time in order to obtain a feasible solution. The distance and route time are factored into the cost function as they indirectly affect time violation constraints. The total route cost is the sum total of each vehicles route cost.

During the genetic sectoring process, a set of seed angles are represented in a chromosome with each seed angle occupying 3 bits of the chromosome. Figures 10 and 11 show the represenatation and the results of a 2-point crossover operaton. The binary representation of each seed angle, B_i, is converted to the integer format using the following equation:

$S_i = (\frac{Max-Angle}{Fleet-Size}) + INT (B_i) * C$, where C = 1.5 Max-Angle is the maximum angle value of the customers in the problem and INT is a function that converts a binary string to an integer value. The fitness value of a chromosome is the total route cost of the clusters formed from the set of seed angles in the chromosome.

Once the clusters are formed, the routing of a cluster begins with vehicle from the depot randomly selecting a customer from the cluster as the first to be visited. Assume that the depot is denoted by 0 and the first customer selected to be served is 3, giving an initial route 0-3. If the next customer to be served is 5, the customer can be inserted between the depot, 0, and customer 3, to form the route 0-5-3 or after customer 3 to form the route 0-3-5. The insertion location of customer 5 into the route 0-3 that yields the lowest route cost, using cost function (3.2), will be selected as the next insertion point. For a route that has L customers, to insert the next customer, the cost of L+1 locations are calculated and the location with the lowest route cost is chosen as the next insertion point.

3.1.2 The LOCOPT Module

Though the GENSECT module is effective in searching the entire search space for the set of seed angles with the lowest total route cost, it could still have tardy customers or vehicles that exceed vehicle capacity or travel time. This inherent disadvantage can be removed by switching customers between clusters by applying local optimization procedures to the solution from GENSECT. The customer switching method is similar to the cyclic transfer local optimization method of Thompson [88], based on moving customers between clusters to minimize the total route cost. The iteration between GENSECT and LOCOPT can be executed any number of times, but we have found that 2 iterations are sufficient to attain good solutions.

3.2 Computational Results

NGO-VRPTW was run on a set of 56 problems, in six data sets, developed by Solomon[83]. The problems were also solved by Thomson[88]. NGO-VRPTW did better than the two alternatives on 41 of the 56 problems with an average reduction of 3.9% in fleet size and 4.4% in distance traveled by the vehicles. Each problem took an average of 127 cpu seconds to be solved on a SOLBOURNE 5/802 machine. Each of the problems in these data sets has 100 customers and the fleet size to service them varies from 2 to 21 vehicles. They incorporate many distinguishing

features of vehicle routing with two sided time windows. The problems vary in fleet size, vehicle capacity, travel time of vehicles, spatial and temporal distribution of customers, time window density (the number of demands with time windows), time window width and customer service times. The geographical distribution of the customers consists of randomly generated customers, in data sets R1 and R2, clustered customers, in data sets C1 and C2, and semi-clustered customers, in data sets RC1 and RC2. The problem sets R1, C1 and RC1 have small vehicle capacities and travel time compared to those of R2, C2 and RC2. Solutions to each problem were obtained by Solomon[87] and Thompson[88]. Solomon tested a number of algorithms and heuristics and reported that the overall best performance was obtained using the "I1" insertion procedure. Solomon[83] reports the best solutions using the "I1" insertion method with eight different combinations of parameters and initialization criteria. Thompson's[88] solutions use local optimization procedures to improve feasible solutions obtained using the "I1" insertion method using the best of eight different combinations of parameters and two different initialization criteria.

For each problem, the results of NGO-VRPTW were the best of two solutions generated by two different initial placement of customers. In one, initial placement the customers were sorted in ascending order of their angles to distribute the customers based upon their geographical locations   In the other, the customer angles were left unsorted. The solutions obtained by NGO-VRPTW were compared against those of Solomon, denoted by Method 1, and Thompson, denoted by Method 2, in Table 1. In the tables, each problem number, PROB, shows the percent differences in fleet size(NV%) and distance(DIST%) of the NGO-VRPTW solution against the other two methods. Positive values indicate improvement over the other two methods. The reduction of fleet size and distance obtained by NGO-VRPT   were comparatively larger for problem sets R1, R2, RC1 and RC2 than for problem ts C1 and C2. Due to the close geographical proximity of customers in problems sets C1 and C2, the genetic sectoring process converges upon a feasible solution relatively quickly.

For randomly located customers the time violations and infeasibilities are greater during the genetic sectoring process, leading to a more rigorous adaptive search and better solutions.

The adaptive nature of the genetic algorithms are exploited by NGO-VRPTW to attain solutions that are superior to those of competing methods. This methodology is potentially useful for solving VRPTW's in real time for dynamically changing customer demands.

Table 1. Comparison of average % difference in fleet size and distance between   the other two methods and NGO-VRPTW over all data sets.

| PROB	METHOD 1		METHOD 2	
	NV%	DIST%	NV%	DIST%
R1	3.8	9.0	2.1	3.3
C1	0.0	4.7	0.0	2.0
RC1	7.0	6.9	3.0	3.4
R2	2.2	19.5	-4.5	10.9
C2	1.2	-8.8	0.0	-16.3
RC2	11.9	15.9	7.1	15.9

Legend:

METHOD 1	:	Data from Solomon's method [Solomon, 1983]
METHOD 2	:	Data from Thompson's method [Thompson, 1989]
PROB	:	Problem set
NV	:	Percent average of fleet size
DIST	:	Percent average of total distance traveled

	3	6	9	12	15	18	21	24	27

Parent 1 Chromosome

1	0	1	1	0	1	0	1	1	1	1	0	0	1	1	1	0	1	1	0	0	1	1	1	1	0	1

Seed 2 (37.5)	Seed 3 (67.5)	Seed 4 (94.5)	Seed 5 (129.0)	Seed 6 (151.5)	Seed 7 (187.5)	Seed 8 (216.0)	Seed 9 (250.5)	Seed 10 (280.5)

	3	6	9	12	15	18	21	24	27

Parent 2 Chromosome

0	0	1	1	0	1	1	0	1	0	0	1	1	1	0	1	1	1	0	1	1	0	0	0	1	1	1

Seed 2 (31.5)	Seed 3 (67.5)	Seed 4 (97.5)	Seed 5 (121.5)	Seed 6 (159.0)	Seed 7 (190.5)	Seed 8 (217.5)	Seed 9 (240.0)	Seed 10 (280.5)
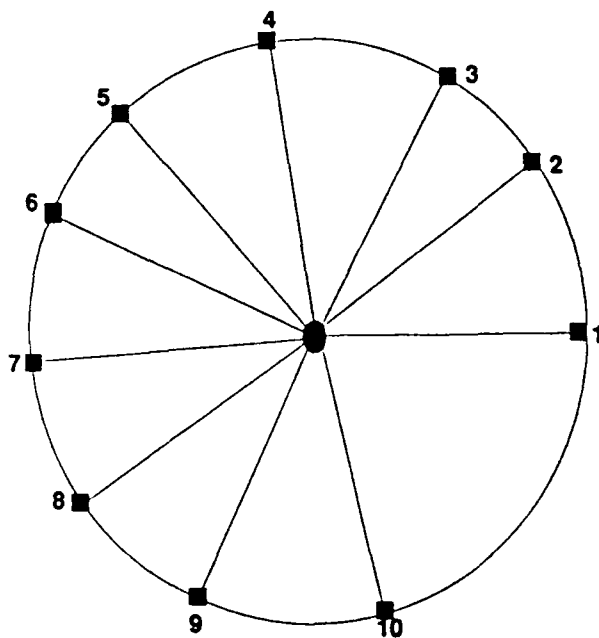
Seed Angles derived from
Parent 1 Chromosome
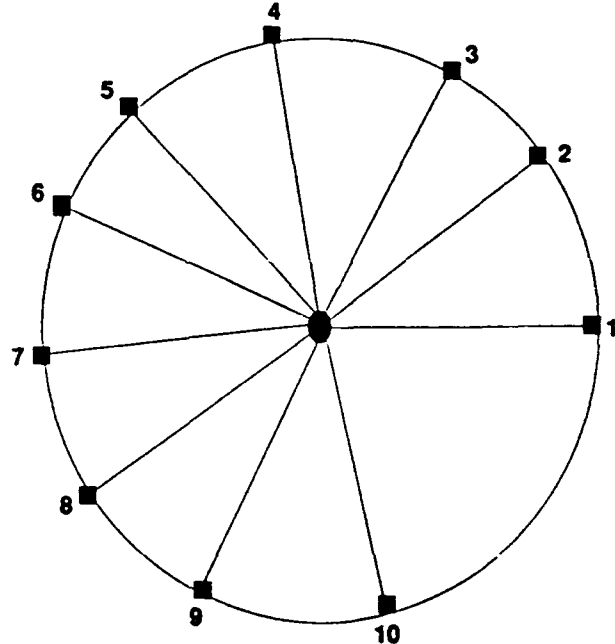
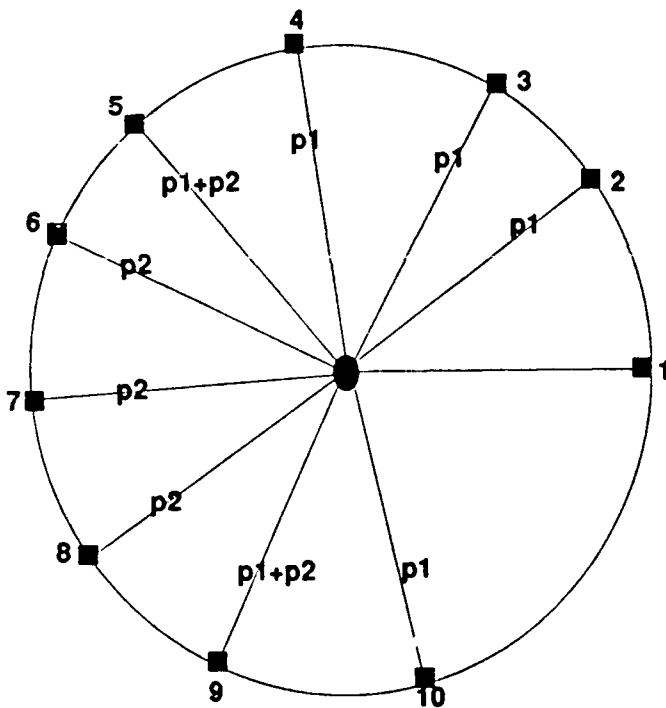Seed Angles Derived from
Parent 2 Chromosome

Figure 10. Representation used for seed angles in the time-constrained problem. Each angle is a variation from a normalized positon.
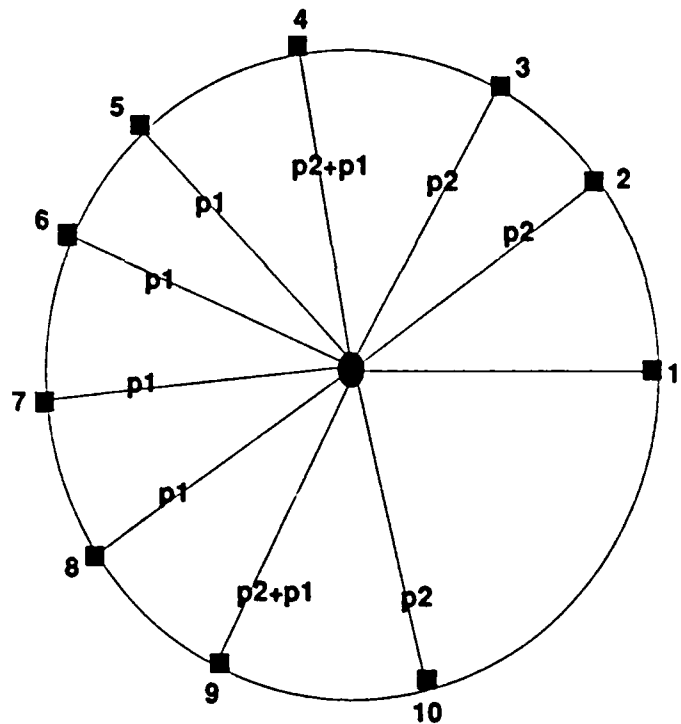
Offspring 1 Chromosome

		3		6		9		12		15		18		21		24		27
1				0	1	1	1	1	1	1	1	0	1	1	1	0	1	1	0	1	1	1	0	1

Seed 2 (37.5)	Seed 3 (67.5)	Seed 4 (94.5)	Seed 5 (130.5)	Seed 6 (159.0)	Seed 7 (190.0)	Seed 8 (214.0)	Seed 9 (249.0)	Seed 10 (280.5)

Offspring 2 Chromosome

		3		6		9		12		15		18		21		24		27
0	0	1	1	0	1	1	0	1	0	0	0	0	1	1	1	0	1	1	0	0	1	0	0	1	1	1

Seed 2 (31.5)	Seed 3 (67.5)	Seed 4 (97.5)	Seed 5 (120.0)	Seed 6 (151.0)	Seed 7 (187.0)	Seed 8 (216.0)	Seed 9 (246.0)	Seed 10 (280.5)

Seed Angles derived from
Offspring 1 Chromosome

Seed Angles Derived from
Offspring 2 Chromosome

Figure 11. Example offspring after 2-point crossover is performed on the parents in Figure 10. Only the angles at the crossover points are modified, propagating high performance structures in the remainder.

4.0 The Remote Autonomous Vehicle System

Preliminary steps have been taken to apply the methodology to a problem of fundamental military importance, the routing and mission planning of autonomous vehicles. A primary example is the Tomahawk cruise missile. Such missiles fly pre-programming routes and have the capability of matching terrain data obtained by sensors with representations of the terrain stored in an onboard computer. The work described here has been primarily supported by the Naval Surface Warfare Center, Dahlgren, Va, but the basic research activities conducted for DARPA under the present project contribute to progress on this application. We refer to the system under development as the adaptive mission operation scheduler (AVOS). A primary goal of the research is to provide an adaptive capability, so that an enroute missile would have the capability of altering course in response to changes in information, such as a new target location or threat region. The methodology under investigation is ideal for this type of application, because of the inherently adaptive nature of the search processes in use. We address first the fundamental problem of determining good trajectories for a remote autonomous vehicle (RAV) to follow from a launch point to a destination point through an hostile environment. The RAV must operate under implicit and explicit constraints. The implicit constraints of the RAV concern fuel capacity, turning radius and other maneuvering limitations. External constraints concern the need to fly over navigation points to re-initialize the guidance systems, reach targets within a time-window, etc. In our approach, we treat reaching a target as the primary goal, and avoiding hostile areas as a secondary goal.

To discretize the system, we superimpose a grid upon the region through which the RAV must fly, and restrict movement to the resolution of the grid points. In our experiments thus far, we use a 45*50 grid. As an approximation of reality, each point on the grid has a hostility measure between 0 and 1. These measures are abstract fuzzy set membership grades. Thus, a hostile grid point of measure .8 mean that the point is associated with membership in the set of hostile grid points with a degree of membership of .8. The values are only loosely associated with probabilities of detection or destruction of the missile. The flow of the algorithm in the AVOS system is depicted in Figure 12.
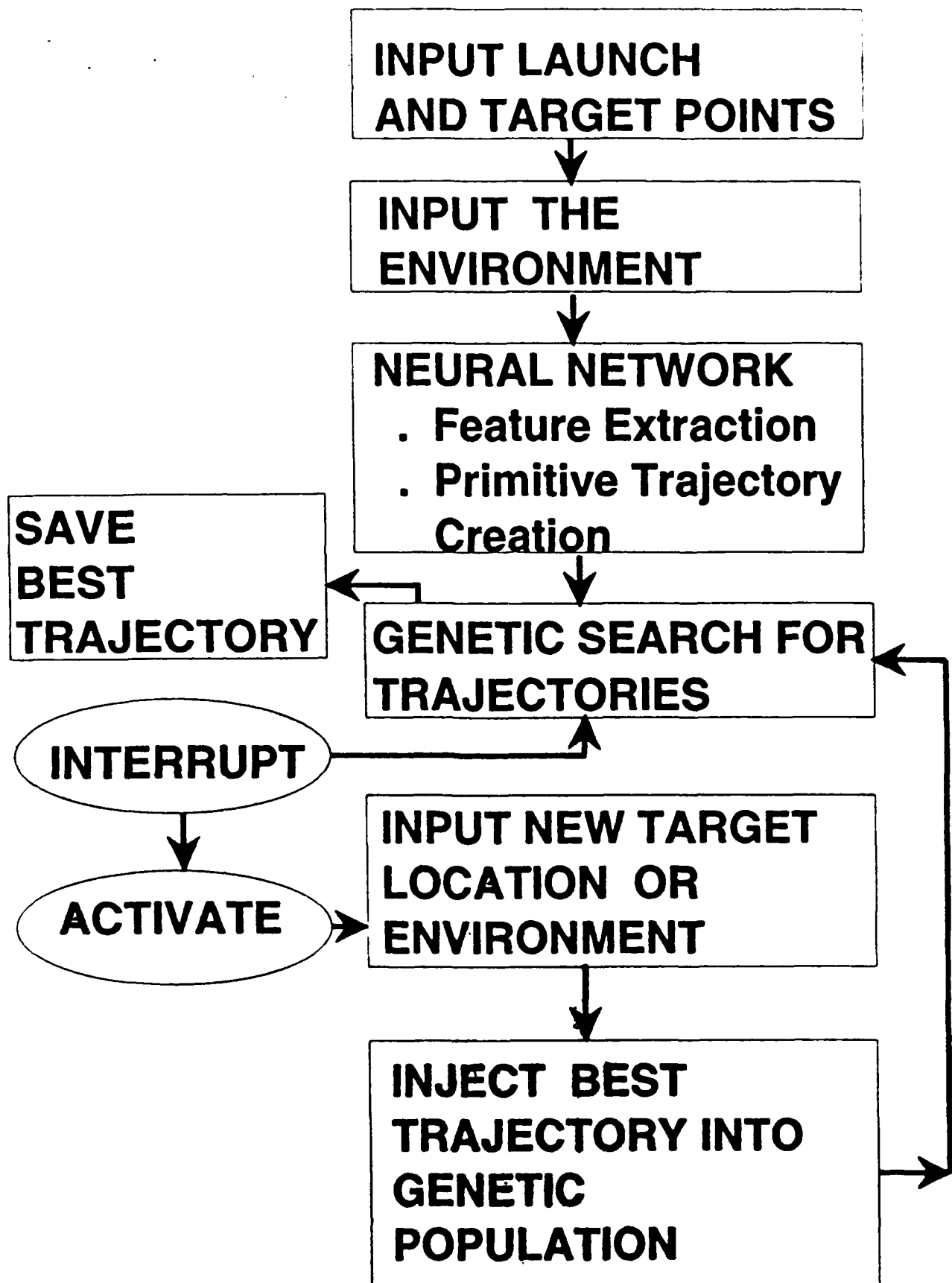
Figure 12. Flow of the procedure for routing an autonomous vehicle. An interrupt with new information can occur at any time.

When the AVOS system is invoked, the launch point, target point and hostile environment information are input into the algorithm. The neural net system is charged with calculation of the parameters for what we call trajectory primitives, used to help initialize the population for the genetic search. These primitives are the parameters for portions of line segments, ellipses parabola and hyperbolic shapes that provide fundamental models for the missile trajectories. The neural net system must extract geographical features of the hostile region as well as launch point and target site relationships, and arrive at a rudimentary pattern for the missile to follow. This aspect of the problem is consistent with the success of neural networks in shape recognition and pattern classification. Our work on this portion is in preliminary stages, but already we have been able to produce reasonable fits of several primitives to the data in several problems.

After initialization, a genetic search seeks a detailed trajectory that the missile begins following. An interrupt can be invoked at any point after the missile is launched to change the conditions of the environment or the location of the target. After an interrupt, the genetic algorithm is invoked to plot a new trajectory from the current point of the missile to the target, through the new environment, the genetic algorithm uses a the previous trajectory as a guide to adaptively plot a new trajectory. Thus far, a simple representation designed to illustrate the adaptive capability of the methodology is in use. In this representation, shown in Figure 13, we employ a chromosome that is partitioned into 3-bit portions, allowing 8 distinct integers to be represented by such groups. The 8 integers correspond to the 8 nearest neighbors in the grid, for a 2-dimensional search. Figure 14 shows convergence of the search for a typical problem from a random start. In the example, there is navigation point represented by a triangle, the small square is the launch point, and the larger square is the target. Areas of hostility at a single level are shown as dark squares. Note that early on in the search, the solutions are being primarily rewarded for reaching the target, with secondary importance for avoiding hostility. Ultimately, the search finds a trajectory that flys through very little hostility, flys sufficiently close to the navigation point, and also reaches the target. The final solution is smoothed by a b-spline to maintain adherence to internal constraints. The length of the artificial chromosomes models the fuel limitation of the missile. The type of trajectory generated is dependent on a fitness function that measures the desirability of the trajectory. The cost function used in evaluating the fitness of the target calculates the the difference in distance between the target and the point in the trajectory that is closest to it, the navigation point and the point in the trajectory that distance s closest to it and the number of hostile points through which the trajectory passes. The cost function is defined as follows: $W_d D_i + W_n N_i + W_t T_i + \sum_{i=1}^{n} W_h h_i f_i$ where:

W_D —   weight factor for the distance away from target.

D_1 —   difference $\in$ distance between the point $\in$ the trajectory
    closest to the target and the location of the target

W_n —   weight factor for the distance away from the
   navigation point.

D_2 —   difference $\in$ the distance between point $\in$ the trajectory
    closest to the navigation point and the location
       of the navigation point.

W_t —   weight factor for the time taken reach target.

D_3 —   number of units required to reach target.

W_h —   weight factor for hostility.

h_i —   presence or absence of a hostility at a grid location i.

f_i —   probability of hostility at point i.

In the experiments thus far,
the weight factors were set as $W_D = 1000$, $W_n = 50$, $W_t = 50$ and $W_h = 50$.

The AVOS system is designed to have the capability of dynamically
responding to changes in the

environment. Figure 15 shows adaptation to a change in target location. In this illustration, the missile was underway and near the navigation point when it was learned that the target location was actually in the hostile region, some consideiable distance up from the old target in the Figure. Note that the search underway was able to respond to the change, and identified a trajectory that reaches the modified location. In other experiments, the locations of threat points in the hostile region were modified, and also simultaneous modification of both threat region and target location. The methodology successfully adapts to these changes. Computation is relatively fast. The trajectories are typically generated in about 5 cpu seconds in an SOLBOURNE 5/802. Although much remains to be done, the research shows that the approach has potential to provide good trajectories that could be adaptive in real time.
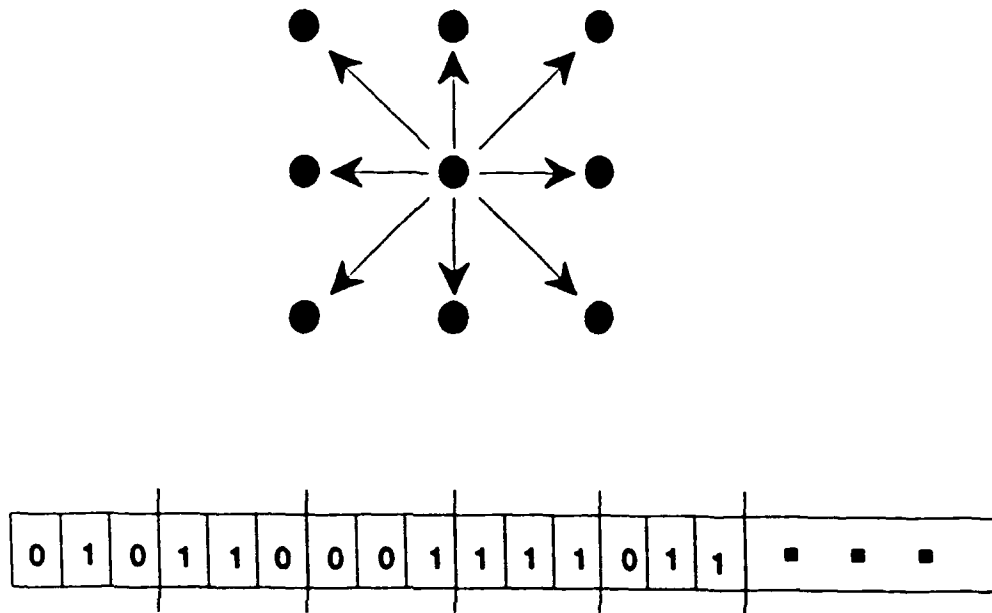
Figure 13. Representation as an artificial chromosome of the directions of turn available to the vehicle.
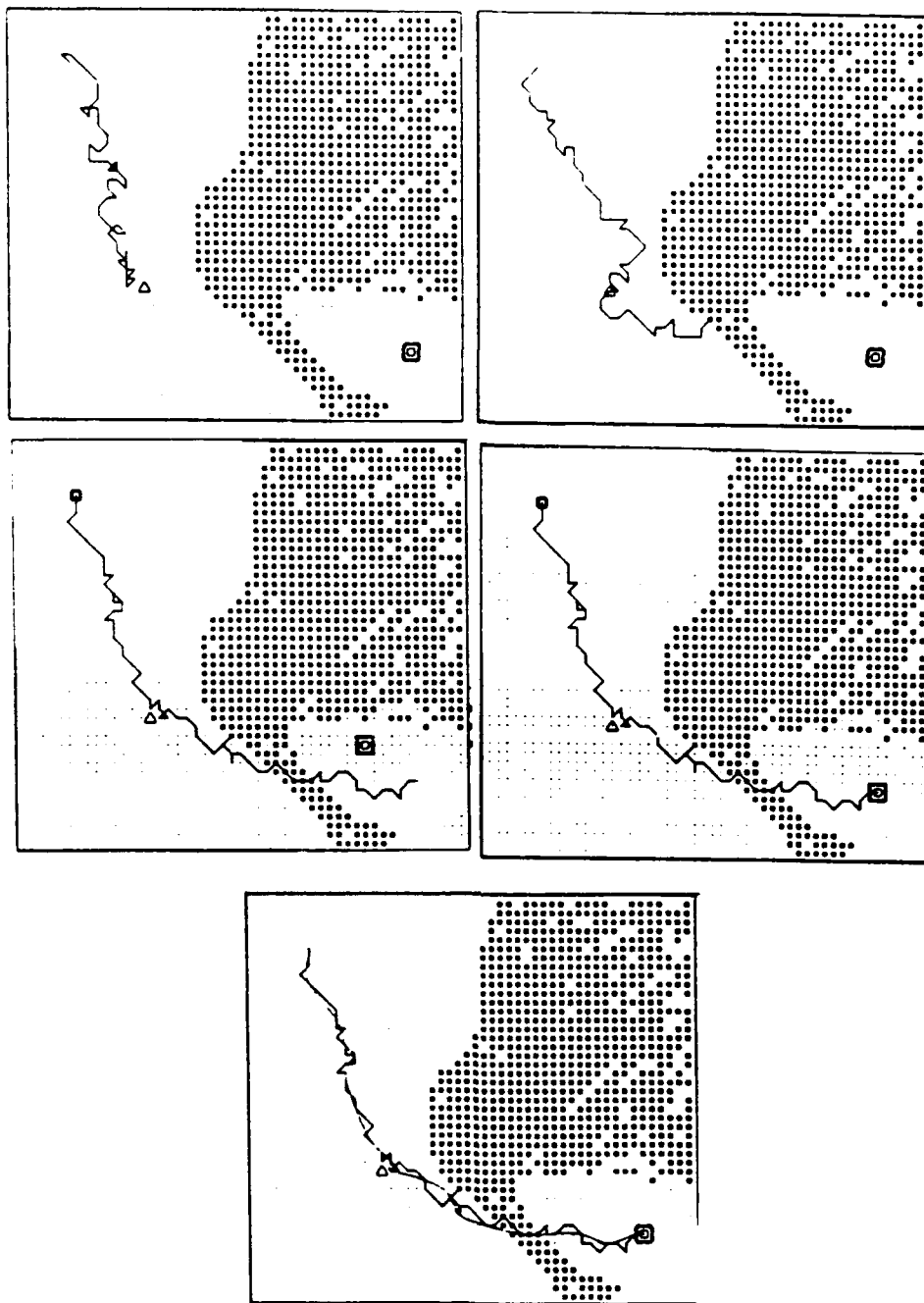
Figure 14. Convergence to a high performance trajectory from a random start. The trajectory successfully finds both the navigation point and the target, and also minimizes exposure to threat by evading hazardous territory.
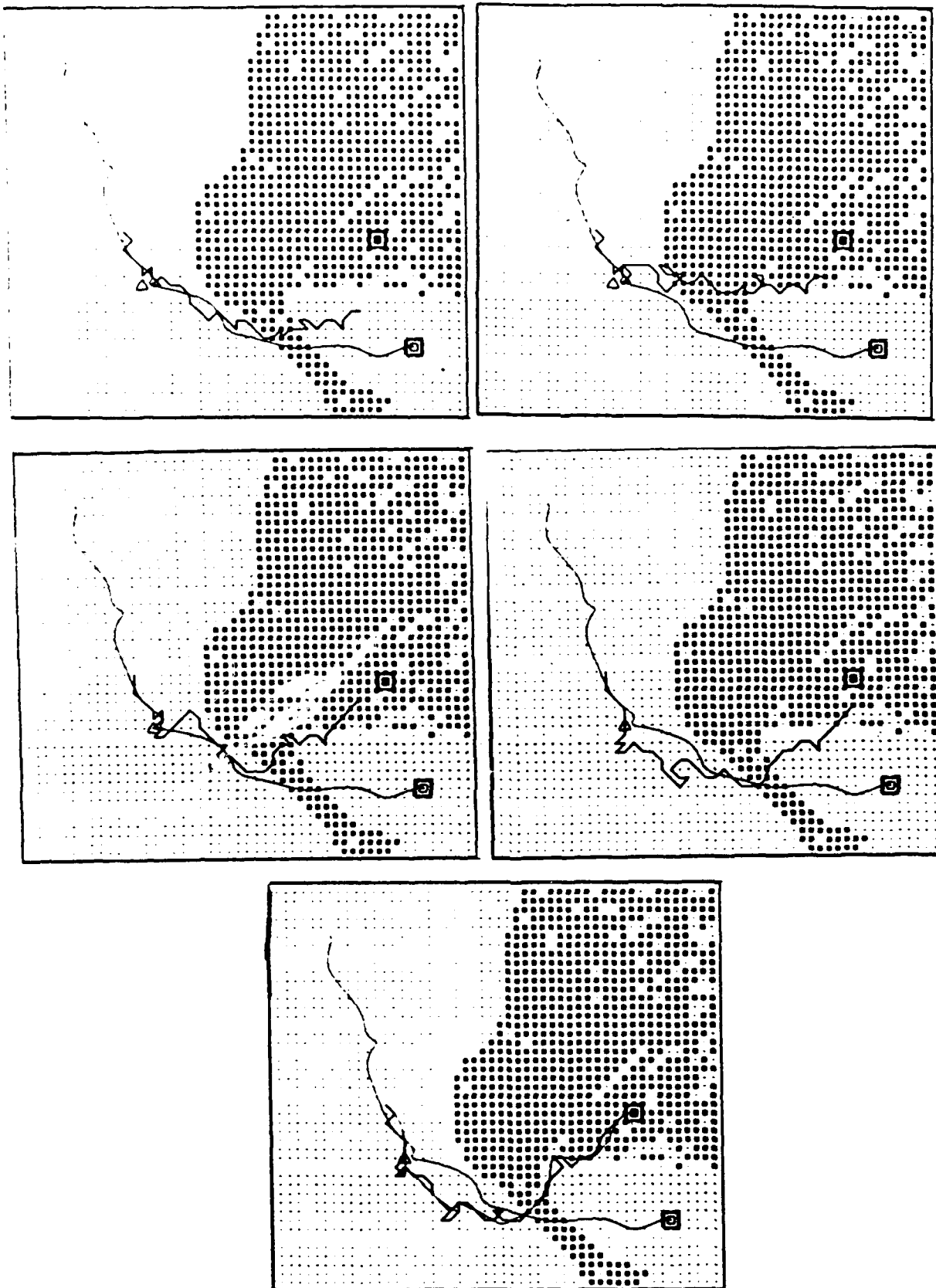
Figure 15. Adaptive capability of the procedure. Instructions to seek a new target location are received enroute, and the trajectory is dynamically modified to find the new location.

5.0 Future Research Directions

Most research in vehicle routing and scheduling has focused on static models. However, in practice, routing problems are extremely dynamic. Consider, for example, breakdowns within a vehicle fleet, loads that were expected that did not materialize after a vehcle was already underway, desirable new loads that suddenly become available, unexpected delays or accelerations elapsed time, etc. In military applications, such as the autonomous vehicle problem, targets may be reached by other weapons systems, threat areas can suddenly change, sensor and satellite data can reveal new information, or mission priorities can change, all making it desirable to dynamically reroute vehicles. In addition, sophisticated 2-way communications are rapidly making it economical and feasible to dynamically track and control vehicles anywhere in the world, opening new possibilities for more increasing their efficiency. The approach that we pursue has considerable potential for addressing dynamic routing problems, because of the inherent adaptive nature of the methodology. In addition, for mission planning purposes, many weapons systems are employed during overlapping time frames. Our results on generic routing problems with multiple vehicles strongly suggest that there is considerable potential to extend these methods into systems for helping mission planning and deployment decisions make optimal decisions in utilizing a multiplicity of smart weapons. We are also making efforts to develop appropriate graphical user interfaces for routing models, so that the technical information concerning routing decisons can be readily presented to human operators responsible for dispatching decisions. This area has strong interconnections with image processing, (especially satellite imagery), Geographical Information Systems (GIS), and general decision support systems.

REFERENCES

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

Baker, Edward K. and Joanne R. Schaffer (1986), *Solution Improvement Heuristics for the Vehicle Routing Problem with Time Window Constraints*, American Journal of Mathematical and Management Sciences (Special Issue) 6 (3-4), 261-300.

Baker, J. E. (1985), *Adaptive Selection Methods for Genetic Algorithms*, Proceedings of the First International Conference on Genetic Algorithms and their Applications, Pittsburgh, 101-111.

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 (Special Issue) 10 (2), 63-211.

Clarke, G. and Wright, J. (1964), *Scheduling of Vehicles from a Central Depot to a Number of Delivery Points*, Operations Research 12 (4), 568-581.

Fisher, Marshall L. and Ramachandran Jaikumar (1981), *A Generalized Assignment Heuristic for Vehicle Routing*, NETWORKS 11, 109-124.

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.

Goldberg, D., (1989), *Sizing Population for Serial and Parallel Genetic Algorithms* of the Third International Conference on Genetic Algorithms and their Applications, George Mason University, 70-71.

Goldberg, D. (1987), *Genetic Algorithms with Sharing for Multimodal Function Optimization*, Proceedings of the Second International Conference on Genetic Algorithms and their Applications, MIT, 41-49.

Golden, B. L. and Stewart, W. R. (1985), *Empirical Analysis of Heuristics* in *The Traveling Salesman Problem*, E. Lawler, J. K. Lenstra, A. Rinooy Kan and Shmoys (Eds.).

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.

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

Jog, Pl, J. Y. Suh, and D. V. Gucht, (1989), *The Effect of Population Size, Heuristic Crossover and Local Improvement on a Genetic Algorithm for the Traveling Salesman Problem*, Proceedings of the Third International Conference on Genetic Algorithms and their Applications, George Mason University, 110-115.

Kadaba, N, K. E. Nygard and P. L. Juell (1990), Integration of Knowledge-Based Systems and Adaptive Learning Techniqes for Routing and Scheduling Applications, Expert Systems with Applications.

Lenstra, J. and Rinnooy Kan (1981), *Complexity of the Vehicle Routing and Scheduling Problems*, NETWORKS 11 (2), 221-228.

Nygard, K. E., and H. Liu (1988), *A Spanning Tree Algorithm for Routing*, Technical Report, Department of Computer Science and Operations Research, North Dakota State University, Fargo.

Nygard, K. E., and R. Walker (1988), Vehicle Routing Algorithm Based on Spacefilling Curves, Technical Report, Department of Computer Science and Operations Research, North Dakota State University, Fargo.

Nygard, K. E., and C. H. Yang (1990), A Statistical System for Comparing Alternative Routing Algorithms, Technical Report, Department of Computer Science and Operations Research, North Dakota State University, Fargo, North Dakota. (Also presented at the ORSA/TIMS joint national

meetings, Philadelphia, October, 1990.)

Savelsbergh M. W. P. (1985), *Local Search for Routing Problems with Time Windows*, **Annals of Operations Research 4**, 285-305.

Schaffer, D., R. A. Caruana, L. J. Eshelman, and R. Das (1989), *Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization*, Proceedings of the Third International Conference on Genetic Algorithms, George Mason University, 51-60.

Solomon, Marius M., Edward K. Baker, and Joanne R. Schaffer (1988), *Vehicle Routing and Scheduling Problems with Time Window Constraints: Efficient Implementations of Solution Improvement Procedures*, in *Vehicle Routing: Methods and Studies*, B. L. Golden and A. A. Assad (Eds.), Elsevier Science Publishers B. V. (North-Holland), 85-90.

Solomon, Marius M. and Jaques Desrosiers (1988), *Time Window Constrained Routing and Scheduling Problems: A Survey*, **Transportation Science 22** (1), 1-11.

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

Solomon, Marius M. (1983), *The Vehicle Routing and Scheduling Problems with Time Window Constraints*, Ph.D. Dissertation, Department of Decision Sciences, University of Pennsylavania.

Thangiah, Sam R. (1991), *GIDEON: A Genetic Algorithm System for Vehicle Routing with Time Windows*, PhD Dissertation, North Dakota State University, Fargo, North Dakota.

Thompson, Paul M. (1988), *Local Search Algorithms for Vehicle Routing and Other Combinatorial Problems*, Ph.D. Dissertation, Massachusetts Institute of Technology, Massachusetts.