Proceedings of the 2008 Winter Simulation Conference
S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler eds.

Multi-Objective UAV Mission Planning Using Evolutionary Computation

Adam J. Pohl
Gary B. Lamont

Department of Electrical and Computer Engineering
Graduate School of Engineering and Management
Air Force Institute of Technology
WPAFB (Dayton), OH 45433-7765, U.S.A.

Abstract

This investigation develops an innovative algorithm for multiple autonomous unmanned aerial vehicle (UAV) mission routing. The concept of a UAV Swarm Routing Problem (SRP) as a new combinatorics problem, is developed as a variant of the Vehicle Routing Problem with Time Windows (VRPTW). Solutions of SRP problem model result in route assignments per vehicle that successfully track to all targets, on time, within distance constraints. A complexity analysis and multi-objective formulation of the VRPTW indicates the necessity of a stochastic solution approach leading to a multi-objective evolutionary algorithm. A full problem definition of the SRP as well as a multi-objective formulation parallels that of the VRPTW method. Benchmark problems for the VRPTW are modified in order to create SRP benchmarks. The solutions show the SRP solutions are comparable or better than the same VRPTW solutions, while also representing a more realistic UAV swarm routing solution.

1 INTRODUCTION

This paper proposes a new problem model for the development of unmanned aerial vehicle (UAV) routing solutions via a study of the routing of multiple UAVs and UAV swarms to a set of locations while meeting constraints of time on target, total mission time, enemy radar avoidance, and total path cost optimization. Contemporary Research is focused, and increasingly continues to focus, on the development of autonomous self-organized UAVs. Developing a single autonomous UAV is not the objective, rather the objective is to develop a massive array of autonomous UAVs, capable of working together toward a common goal. The term for this array is a swarm or flock for which there are many different design approaches as the problem itself exists in many scientific and engineering domains. This research focuses on the development of off-line UAV routing and mission planning, combined with a simulation and visualization system the purpose of which is to better understand the computational complexity of autonomous UAV swarm routing.

2 PROBLEM FORMULATION

The problem of mission planning consists of assigning multiple vehicles sets of targets to visit. These targets exist in a field of uneven terrain where different enemy radar lines of sight exist. There exist two problem aspects to deal with, the first is the development of flight paths between targets. The path must be optimized for cost and risk. Cost is how much energy or time it takes to traverse the path and risk is a measure of how dangerous the flight area is. The second is the development of path order. Once it is determined how to best fly between targets the order of these flight paths must be determined. Generating the cost of the path is a separate and immaterial problem related to the development of path order. In fact, the development of single path optimization is already being studied in fields such as robotics, land, and air based agents. Once these path costs are known, or estimated, however what development process should be used to structure a valid route plan from them? This process of route development is the subject matter of this investigation.

In order to model this routing situation, a combinatorics problem known as the VRPTW is used. The VRPTW encompass a situational problem composed of a number of vehicles, known targets with time visitation constraints, and constraint on the visitation capacity of each vehicle. This model most efficiently possesses all the aspects of the problem under consideration and is well documented and understood. The VRPTW is limited in its ability to model realistic UAV routing, necessitating in its extension into a new problem model. This innovative VRPTW variation is called the swarm routing problem, SRP, which presented as a more efficient form to model the routing of multiple UAVs to multiple targets concurrently (Pohl 2008).

978-1-4244-2708-6/08/$25.00 ©2008 IEEE

2.1 Vehicle Routing Problem with Time Windows

The VRP is a well established combinatorics problem with many variations and solutions, one of these variations being the VRPTW (Toth and Vigo 2001). The VRPTW consists of a set of targets, some number of vehicles, and a depot. The depot is the deployment and return point for all the vehicles. Each target (and the depot) has a Euclidian location (coordinate), some associated demand (except the depot), an arrival time window, and a path to every other location. The objective of the problem is to develop a set of routes for each vehicle, so that all targets are visited within the time window, the associated demand is met, and all vehicles return to the depot on time. Each vehicle in the problem has a set capacity that it can not exceed while visiting customers. Visiting a customer subtracts its demand from the vehicles capacity. The total demand on the vehicle is the sum of the demands of all the customers visited. If the vehicles being used do not have some type of capacity constraint the problem then decomposes to a TSP since one vehicle can now satisfy all customers.

The VRPTW, as formulated by (Toth and Vigo 2001) based on the ordinal formulation by Solomon (Solomon 1987), is defined by a fully connected graph where each edge has associated with it some travel cost. A mathematical model formulation of the single-objective VRPTW based upon the above nomenclature is found in (Toth and Vigo 2001). This model provides the mathematical foundation for the SRP model.

2.2 Multi-objective Evolutionary Algorithms

Evolutionary algorithms are capable of providing polynomial time ”acceptable” solutions for many NP and NP-Complete problems that would otherwise require an exponential or intractable ”optimal” solution time. One of the interesting aspects of NP problems is that many of them can be extend as multi-objective problems. There are two conflicting effects of a multi-objective problem. The first is that the problem is often more useful because it more closely approximates reality, but this comes at a cost of the second effect: increased complexity. The VRP is hardly realistic, however the VRPTW described earlier in this section is closer to reality, and if more constraints are applied, such as heterogenous vehicles, back-hauls (pick up and delivery), or multiple depots, the problem would become even more realistic. While this makes the solution of the problem more valuable it also makes an optimal solution that much more difficult to obtain.

With knowledge of the effective use of evolutionary algorithms and the need for multi-objective problem solutions, recent work has focused on developing multi-objective evolutionary algorithms (Coello Coello 2007). MOEAs are basically the same as a standard single objective GA with the difference of how multi-objective solutions are evaluated and ranked. Note the concept of dominance in a multi-objective solutions. Thus, a solution is said to dominate if there is no other solution that can improve one of the objectives without simultaneously reducing another.

By examining the dominance of different solutions and ranking them accordingly, a more accurate selection of effective solutions can be made for future generations. Also, by ranking across multiple objectives the resulting solution achieves optimal performance across all objectives without being biased toward any one objective. When discussing optimality in a multidimensional space the concept of the pareto front becomes beneficial. The Pareto front is the set of non-dominated, feasible solutions. More recent MOEAs use this understanding of the Pareto front in order to track which genotypes are developing better solutions. The resultant set of solutions in the front provides solutions with different tradeoff values. Which solution is actually used is a decision made by a user or by some pre-determined rule.

An examination of the current literature reveals a growing appreciation for the use of MOEAs in complex problems, such as the VRP (Lou and Shi 2006). It is shown that MOEAs are better able to navigate the highly irregular solution space that exists within the VRPTW. What has also been shown is that multi-objective approaches not only develop good solutions but are also better than biased single objective solutions for the optimization of any of a problems multiple objectives (Ombuki, Ross, and Hanshar 2006) for certain problems.

2.3 The Swarm Routing Problem

In almost all variations of the VRP, or VRPTW, it is assumed that all vehicles depart from the depot location to different targets and only one vehicle visits each target (Toth and Vigo 2001). This problem model is appropriate because each vehicle is generally assumed to be ground based. However, the use of a UAV swarm introduces an interesting aspect which, up to this point, has not been dealt with. When dealing with a swarm of UAVs and multiple targets to visit, it is desirable to have the ability to route the swarm between targets in the most efficient manner possible. The reason for this is that often many targets exist on a battlefield which need to be visited in a timely manner, while also utilizing resources in the most economical fashion possible. It would be much more efficient to be able to consider UAVs as a dynamic group rather than indivisible units that can only visit one location at a time. This implies an imperative to take advantage of the divisibility of the swarm and route subgroups of UAVs to different targets, as it is deemed efficient to do so, and then regroup at other targets.

By viewing the problem in this manner the value of importance changes from edge costs between targets to the distance traveled by each UAV. Within the VRPTW (and

VRPs in general) each vehicle is seen to have some capacity associated with it that is used to satisfy each customer. While this works for ground based delivery routing, it would be more appropriate to view target satisfaction as the number of UAVs on target in some time window. The path cost associated with a single vehicle is then more reflective of the distance that needs to be traveled, and the use of many vehicles capable of being routed through multiple targets (causing splits and joins in the group in the process) presents a more realistic and useful problem model.

The SRP problem domain consists of a network $G = (V, A)$ where $V = \{v_0 v_1, \ldots v_n\}$ and v_0 is the depot. The set of edges is defined as, $A = \{(v_i, v_j) \in V, i \neq j\}$, where each edge has associated with it some cost $c(v_i, v_j)$. The cost of the edge is the cost of travel from target i to target j. For now we assume path cost as some constant travel speed for each UAV making the travel cost simply the Euclidean distance between points.

A time window exists for all customers where, E, represents the earliest start time and, L, the latest arrival time. The latest arrival time is the point at which the UAV can arrive and still complete the service time defined by S. If the vehicle arrives earlier than E, it incurs a waiting time, W, which is the difference between the arrival time and E. The total time a vehicle takes to complete it's route is the summation of all route path travel costs, waiting times, and service times ($\sum c_{ij} + \sum w_i + \sum s_i$). The total path time must not exceed the latest arrival time (i.e. closing time) of the depot.

Each customer also has associated with it some demand, D. The demand is an indication of the number of UAVs that need to be present at the target within its time window for the required service time. *This is one of the key differences between the SRP and VRPTW.* Instead of demand being satisfied by a UAVs capacity, it is satisfied by the number of UAVs at the target. The service time indicates the amount of time the UAVs are required to be on target.

There exists K homogenous UAVs each of which has some travel capacity, F, and unit deliverable capacity. Groups of UAVs are classified as a swarm. A swarm can split into one or more sub-swarms, join with other sub-swarms into a larger swarm, and travel along path edges together as a swarm. It is assumed that join and split operations only occur at targets in order to simplify problem complexity. The travel capacity is not a deliverable value as it has been in previous versions of this problem, it is only an indication of how far the individual UAV can fly. This constraint can be viewed as the UAVs power supply limitation. The deliverable value that satisfies the target is equal to the total number of UAVs present in a given location at a given time. This value fulfills the demand requirement of the target during its service time.

The solution to the problem is basically the same as the VRPTW, a list of ordered targets for each vehicle, such that the visitation to each target fulfills all target needs without violating any time or demand constraints. Note, that the cost of a route is not the total time the route takes to complete, it is only the sum cost of the edges the vehicle traverses. The objective remains the same: determine the set of paths for the UAVs such that the total distance is minimized.

The following is a mathematical model formulation of the SRP, based on the VRPTW model found in (Toth and Vigo 2001). Vehicles are defined within the problem by their inclusion in a flow variable, x_{ijk}, which is a binary value indicating if UAV, k, exists on the path that connects $(i, j) \in V$ at any point in the solution. A time variable, ω_{ik}, indicates the start time of UAV k at location i. The subscript $j \in \Delta^{\pm}(i)$ indicates the set of edges from i to j where j is not equal to i, the plus or minus indicates either a forward or backward move along the path.

A_{ij} - Edge cost between i and j
V_n - Network vertices for n target (v_0 is the depot)
E_n - Earliest arrival time of target n
L_n - Latest arrival time of target n
S_n - Service time of target n
D_n - Demand of target n
K - Set of UAVs
F_k - Travel capacity of UAV k

$$x_{ijk} \in \{0, 1\} \qquad \forall k \in K, (i, j) \in A, \tag{1}$$

$$\omega_{ik} \geq 0 \qquad i \in N, k \in K, \tag{2}$$

Equations (1) and (2) define the flow and time variables used. The flow variable is a binary value that indicates vehicle, k, travels from location, i to j, if equal to one, and zero otherwise. The time variable specifies the start time at location, i, by vehicle k.

$$\sum_{j \in \Delta^+(i)} x_{ijk} = 1 \qquad \forall i \in N, \forall k \in K, \tag{3}$$

$$\sum_{j \in \Delta^+(0)} x_{0jk} = 1 \qquad \forall k \in K, \tag{4}$$

$$\sum_{i \in \Delta^-(n+1)} x_{i,n+1,k} = 1 \qquad \forall k \in K, \tag{5}$$

$$\sum_{i \in \Delta^+(j)} x_{ijk} - \sum_{i \in \Delta^-(j)} x_{ijk} = 0 \qquad \forall k \in K, \forall i \in N, \tag{6}$$

Equations (3)-(6) define the edge constraints of the graph in a solution. They indicate that the each vehicle visits a customer only once (3), that all vehicles must start from the depot (4), that all edge costs are symmetrical (5), and that all vehicles must return to the depot (6). Note, in Equation (3) the lack of the summation over K which removes the constraint that each customer be visited by only a single vehicle.

$$x_{ijk}(\omega_{ik}+s_i+t_{ij}-\omega_{jk}) \leq 0 \quad \forall k \in K, (i,j) \in A, \tag{7}$$

$$e_i \sum_{j\in\Delta^+(i)} x_{ijk} \leq \omega_{ik} \leq l_i \sum_{j\in\Delta^+(i)} x_{ijk} \quad \forall k \in K, \forall i \in N, \tag{8}$$

$$e_i \leq \omega_{ik} \leq l_i \qquad \forall k \in K, i \in (0, n+1), \tag{9}$$

Equations (7)-(9) define the time constraints of the problem. Equation (7) indicates that the arrival time at location *i* plus the service time and travel time to the next location must equal the arrival time at the next customer. Equation (8) defines the need for arrival times to be within the customers time window. The depot also has a time window associated with it (opening and closing time) which all vehicles must adhere to (9).

$$\sum_{i\in N} d_i \sum_{j\in\Delta^+ i} x_{ijk} \geq \sum_{k=0}^{K} k_i \qquad \forall k \in K, \tag{10}$$

$$\sum_{j\in\Delta^+ i} c_{ij} x_{ijk} \leq F_k \qquad \forall k \in K, \tag{11}$$

Up to this point the formulation is basically been the same as the VRPTW with the exception of Equation (3). Equations (10)-(11) are what separate the SRP from the VRPTW. Equation (10) indicates that the demand of each customer is satisfied by the number of vehicles on location and that that number must be either equal to or greater than the demand of the target. Equation (11) indicates that the total cost of the path for a vehicle not exceed the vehicles flight cost limit.

$$\min \sum_{k\in K} \sum_{(i,j)\in A} c_{ij} x_{ijk} \tag{12}$$

The single objective function is defined by Equation (12) which illustrates the desire to minimize the total path cost for all vehicles. The path cost does not include the service or waiting times. Time is only a constraint that causes some routes to be infeasible, the cost is the total distance traveled.

This formulation has introduced the SRP as a modification of the VRPTW. By changing the constraints of customer visitation and how a customers demand is satisfied the problem becomes a more realistic model for routing UAVs to multiple targets within a time window. Up to this point the VRPTW and SRP have only been single objective formulations. In the next section the problem models are expanded in terms of multiple objectives.

2.4 Multi-Objective Formulation for VRPTW and SRP

In Section 2.1 the VRPTW is defined and in the previous section a variant, the SRP, is defined. The objective functions for these two problems indicates that only path length is of critical interest. Even though path length is a primary objective it is not the only objective that can be optimized for in the solution. Consider the following situations that may occur within a problem:

- **Vehicle exceeds its capacity to serve a route** - when this happens the route can be split into 2 or more routes. The split is made when the customer demand causes the capacity of the vehicle to be exceeded.
- **Vehicle arrives early to a customer** - when this happens the service time for the customer is increased by the time spent waiting
- **Vehicle violates a time window by arriving late** - a new vehicle and route are added, splitting the route as described when capacity constraints are encountered.

From these situations we see that the solution to alleviating capacity and time violations is to increase the number of vehicles (and routes). Increasing the number of vehicles is, of course, regressive to the development of optimal paths lengths. This is due to the introduction of depot travel times for each new route. Every time a new route is added the vehicle must first travel from the depot to a location, making the addition of new vehicle routes generally cause an increase in total path length (though not always). It is therefore, advantageous to define 3 objectives to minimize for: path length, vehicle count and total wait time. By optimizing for these three objectives we seek solutions with complementary aspects, and better solutions overall. The objective functions now consist of the following equations:

$$\min \sum_{k\in K} \sum_{(i,j)\in A} c_{ij} x_{ijk} \tag{13}$$

$$\min \sum_{k=0}^{K} k \tag{14}$$

$$\min \sum_{k \in K} \sum_{(i,j) \in A} t_{ik} x_{ijk} \tag{15}$$

Equation (13) defines the minimization of the path length. Equation (14) defines the minimization of the number of vehicles used. Equation (15) defines the minimization of the wait time where the variable t is defined in equation (16) for the VRPTW and (17) for the SRP. The wait time for the SRP is defined as the difference between a vehicles arrival time and the arrival time of the latest vehicle, if the latest arriving vehicles time is past the earliest arrival time of the customer. The latest arriving time is used because operation on a customer can not begin until all vehicles are present. These objective functions apply to both the VRPTW and SRP.

$$t_{ik} = \left\{ \begin{array}{l} e_i - w_{ik}, \text{ if } E > w_{ik} \\ 0, \text{ otherwise} \end{array} \right\} \tag{16}$$

$$t_{ik} = \left\{ \begin{array}{l} e_i - w_{ik}, \text{ if } E > w_{ik} \cap E > w_{iu} \\ w_{iu} - w_{ik}, \text{ if } w_{iu} > E \\ 0, \text{ otherwise} \\ \text{where u is the latest arriving vehicle} \end{array} \right\} \tag{17}$$

Optimizing across multiples objectives is done with two purposes in mind. First, to support the idea that a multi-objective formulation is capable of navigating the solution space more effectively than optimizing for only a single objective. Since the objectives complement each other it would seem logical that optimizing over all of them would achieve better results. What is also noteworthy is that by optimizing for these different objectives, solutions with decreased path lengths should be found as opposed to optimizing solely for path length. This idea was first proposed and tested in Ombuki (Ombuki, Ross, and Hanshar 2006) with beneficial results (i.e. benchmark values are not made worse from the multi-objective approach compare to the single objective approach).

The reason multi-objective formulation is more effective is because the problem under consideration has such an irregular solution space. Time constraints introduce irregularities to the Pareto front such that non-dominated solutions become more isolated. This problem is exacerbated as more constraints are applied to the problem such as heterogenous vehicle fleets or pick-up and delivery problems. Only by optimizing across multiple objectives can the solution space be traversed accurately enough to allow the determination of non-dominated solutions. We now proceed with the high level solution design to this fully formed multi-objective problem.

3 ALGORITHM DESIGN

The design objectives for this research are to develop a solution procedure for the multi-objective routing problems described in the previous section. This solution must return information that can be integrated with previous work on UAV path planning (Slear 2006) and simulation (Corner 2004). The end result of this design is a fully developed form of an algorithm to be applied to the VRPTW and SRP whose output is a set of feasible vehicle routes.

A discussion of the problem complexity is performed in order to illustrate design requirements of the algorithm. For a fixed number of vehicles the VRP and VRPTW problems are NP-Complete (Toth and Vigo 2001). The SRP, as an extension of the VRPTW, can likewise be classified as NP-Complete for fixed fleet sizes. The solution space size of the VRP, S, is approximated by Equation (18).

$$S \approx \frac{\exp^{(\pi\sqrt{2n_!/3})}(n-1)!}{8n\sqrt{3}} \tag{18}$$

This equation is found by combining an integer partition distribution generating function and the the complexity of a single n customer routing problem (Pohl 2008). The result of this approximation yields a solution space complexity of $\mathscr{O}(\exp^n n!)$. For the VRPTW the solution space is the same as it contains the same number of total solutions as the VRP minus some constant number of solutions made invalid by time constraints.

3.1 Multi-Objective Genetic Algorithm Design

Multi-objective GAs differ from single objective GAs in how solutions are stratified. In a single objective algorithm, determining the quality of a solution is a simple matter. If the objective is minimization it is a simple compare operation to the lowest value. For multi-objective optimization this process is not as simple. While it is possible to weight objectives in order to obtain a single value associated with the solution this is not an advisable pursuit. Weighted objectives introduce bias because no weighting procedure can accurately treat the different objectives in a manner such that all objectives are optimized effectively (Coello Coello and Lamont 2004). To accurately classify solutions over a multi-objective domain ,a ranking procedure must be used that takes into account not only the value of the solution across the different objectives but also its proximity to other solutions, which is an indication of the value of the information the solution contains.

There are three major components to the EA design: the replacement method, the chromosome structure, and the genetic operators. The replacement method determines which solutions are kept after a new generation is created. Within this step the solutions are ranked and discarded. The selection methods chosen are shown in the context of the complete EA. The sections following the replacement method expand on the chromosome structure, population initiation, and genetic operators. The chromosome structure is a critical step which drives the effectiveness of the entire algorithm and how the different genetic operations function.

Replacement Method: Two replacement methods are shown in the context of the GA they form. Both of these methods rely on non-dominated sorting and objective space distance to rank and select solutions for the next generation. How Pareto ranking and dominance is utilized multi-objective search is discussed in Section 2.2. As arguments can be made for any given selection method for any given problem two algorithms are selected so that their results could be statistically compared. While many different MOEA selection methods exist, NSGA2 and SPEA2 are chosen for their general acceptance within the research community (Coello Coello 2007).

Non-dominating Sorting Genetic Algorithm II: NSGA2 uses an elitist sorting mechanism of the non-dominated points to first organize the solution set. The result of this sorting is a set of solution ranks. The first rank is the hard non-dominated set. Hard non-dominated refers to a point that is not dominated by any other solution, as apposed to soft non-dominated solutions which are only dominated by those points in the first rank. Each decreasing rank is dominated by more points. These points are then compared to each other in order to determine the distribution of points in the current Pareto front and which points contribute best to an exploration of the solution space. This process is called crowding-distance-assignment. By using the crowding distance and ranking procedure the solutions are ordered by how "good" they are. The next generation is then filled with the best solution until the population limit is reached. The complexity of NSGA2 is $\mathcal{O}(MN^2)$ due to the sorting phase of the algorithm.

Strength Pareto Evolutionary Algorithm II: SPEA2 uses a strength ranking procedure to stratify solutions. Each solution is assigned a strength value based on the number of solutions that dominate it. First, it is determined how many points dominate each solution, this is referred to as the fitness value. Then each dominated point is assigned the sum of all the fitness values that dominate it, this value is then called the strength value of the solution. It is this strength value that is used to rank the solution. The strength ranking procedure ensures that while good solutions are kept, solutions that are more isolated (but still dominated) are also kept in order to ensure better exploration of the solution space. After all the solutions are ranked, an environmental selection method reduces the population to a user specified value. This value is referred to as the archive, which is a misleading term. The archive does not actually save any information from one generation to another. Its purpose is to give the user the ability to control how many points should be saved each generation. The run time of the algorithm is dominated largely by the truncation operation. The fitness assignment procedure requires $\mathcal{O}(N^2)$ while the truncation operation is $\mathcal{O}(N^3)$ where N is the number of individuals.

3.2 VRPTW Chromosome Structure

Any chromosome solution used in a VRP must be able to specify how many vehicles are required and which cities must be visited in what order. The solution chromosome defines a genotype, which is a code corresponding to a phenotype which is the actual solution. In terms of total information the genotype does not need to contain redundant or implied information. For example, in the VRP it is implied that a route starts at the depot and ends there. Encoding this information in a chromosome would therefore be a waste of space. There are three ways to accomplish this, others could be formulated but these have been deemed effective through their repeated usage.

A possible solution structure is a bit string where every bit corresponds to an edge in the solution (and every bit is either one or zero indicating whether it is or is not in the solution). This structure is very simple but grows large very quickly and the organization requirement of the VRP lends itself more toward real valued structures anyway. The second structure is a single array of real values representing each target, the order of which indicates the order of visitation. Each route is separated by zeros as shown in Figure 1. This structure is more efficient but still requires the use of separators to indicate where a route begins and ends.

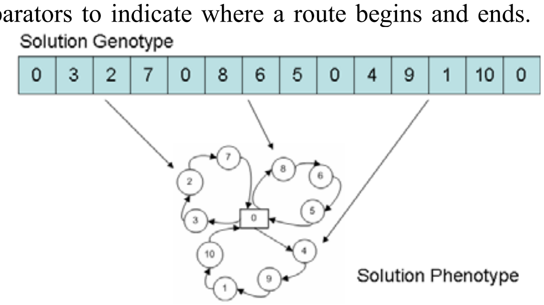

Figure 1: A possible chromosome structure for a VRP.

In (Tavares, Machado, Pereira, and Costa 2003) a structure for a VRP chromosome is defined that uses a similar idea as the array structure but attaches each route to a support structure, like that seen in Figure 2. The most beneficial aspect of this structure is that changes made to a given route do not require a shift to the entire array of values. In (Tavares, Machado, Pereira, and Costa 2003), this structure is proposed, and shown to be, an affective structure especially for the VRPTW.

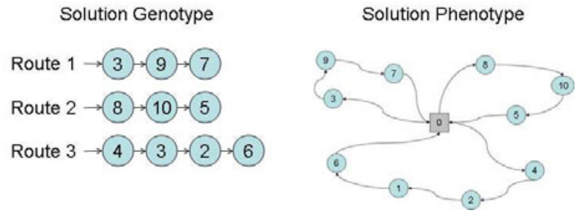

Figure 2: GVR chromosome structure for the VRP.

The GVR structure offers many attributes that make it desirable as a chromosome structure. Its information content does not contain redundancies. Each route implies the existence of a departure and return to the depot even though it is not explicitly stated. This is made possible by the support structure that contains and separates each route. It is also desirable that infeasible solutions are not turned into feasible solutions by adding customers but instead only by rearranging and removing customers. The impact of this is that whenever a solution is checked for feasibility after the addition of a customer it can be safely discarded if infeasible, knowing the solution is a dead end.

3.3 SRP Chromosome Structure

The chromosome structure used for the SRP is essentially the same as for the VRPTW. It consists of single route definitions arranged in a support structure. The only difference is the arrangement of data within the structure. Since each customer must be visited by more than one vehicle at a time the SRP structure must also reflect this. A diagram of this is shown in Figure 3.

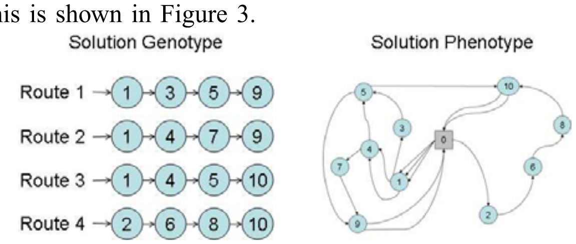

Figure 3: Modified GVR chromosome structure for the SRP.

3.4 VRPTW Genetic Operator Development

The operators described in this section are taken from several publications (Ombuki, Ross, and Hanshar 2006), (Russell and Lamont 2005). The genetic operator alters a solution in a random manner by either randomly changing the solution or optimizing some sub-section of the solution. This opens two avenues to pursue, simple operators applied many times or the use complex heuristics used to intelligently optimize part of a solution. Classic GAs made use of random operators, however more and more hybrid algorithms incorporate heuristics and local search techniques to great effect. For this investigation, a random crossover method, three random mutations, and a heuristic based mutation operator are used.

Random Crossover: Crossover is the genetic operation that occurs most frequently and ensures that children created from the process are feasible. The operation takes two parents, a donor and a receiver. A random selection of customers is selected from the donor and placed into the first available route in a copy of the receiver, after the customers in the incoming sub-route have been removed. The first available route is the route that when receiving the sub-route does not violate any constraints. If no route exists then the sub-route is added as a new route after the copy customers are deleted in the receiver. The net result of this is a child that is a copy of the receiver but contains some sub route section from the donor.

Random Swap Mutation: In swap mutation two random customers in a solution are swapped if doing so does not violate constraints. If constraints become violated the mutation does not proceed. This process is illustrated in Figure 4.

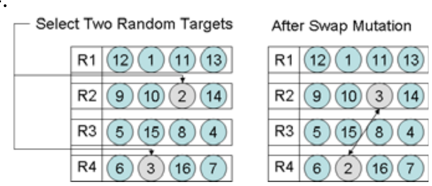

Figure 4: Random swap operator for the VRPTW.

Random Inversion Mutation: Select a random sub-route within a solution and reverse the order of visitation. The mutation does not proceed if this results in an invalid solution. The resultant route may or may not be longer than the original route.

Random Insertion Mutation: Move a random customer to a random location in the solution while ensuring feasibility. It is possible to create a new route with probability $\frac{1}{2V}$ where V is the number of vehicles. (Tavares, Machado, Pereira, and Costa 2003).

Best Route Cost Mutation: This operator randomly selects a route within a solution and optimizes its construction by rearranging customers within that route. This is accomplished by first searching the route and determining which of the customers is closest to the depot. This customer then becomes the first customer. The closest customer to this customer that is in the route is then moved next to the the first customer and so forth, creating a route based on customer proximity. Customers that can not be feasibly added are moved to a new route. It is always assumed that a single customer within a route is valid, without this assumption the problem would not be solvable. The construction of a new route proceeds in the same way, placing customers by order of proximity.

3.5 SRP Evolutionary Operator Development

The SRP genetic operators are variants of the VRPTW operators altered to take into account the different structure of the SRP solutions. The difficultly in developing these operators is ensuring the validity of the child genotype. Since the chromosome contains location sensitive information across two dimensions, as opposed to the VRPTW chromosome which is only sensitive across a single route, making even slight changes can cause invalid solutions to be created.

Split Mutation: Split mutation randomly selects a route within the SRP solution and attempts to reduce the total length of that route be eliminating unnecessary target visitations. Each customer is satisfied with a certain number of UAVs at its location, however more can be present than are actually needed. This may cause a route to be longer than it needs to be since its divergence to an unnecessary target takes longer than a direct route. The split mutation operation determines if this is occurring in a random route and attempts to remove the target from the vehicles flight plan. If this operation then results in an infeasible solution it is considered to have failed, and is not implemented. This process is illustrated in Figure 5.

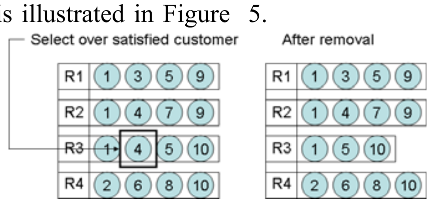

Figure 5: Split mutation operator for the SRP.

Vertical Swap Mutation: The vertical swap operator swaps two different locations vertically in a given solution. This is in contrast to the VRPTW swap mutation in section 3.4 in which the swapped targets can be anywhere. Columns within the SRP have a close approximation to time within the solution. It is not exact because distance information is not contained within the solution, and cities in the same column may not actually be visited at the same time. The swap operator randomly selects a column and two different targets within that column. These targets are then swapped and feasibility is checked. An infeasible solution is not used.

Random Crossover with Tightening: The crossover operation must be done with particular care as effective alterations to the solution are difficult to achieve. The basic idea of the crossover operation is the same as the VRPTW crossover operation, from two solutions a random route is selected from each. This route is then added to the other solution. The problem is, unlike the VRPTW crossover operation, subsections of a route can not be easily transferred between two solutions. In order to compensate for this, the route to be crossed is added to the solution as an entirely new route. The solution then undergoes an operation called tightening. During this operation the solution is searched to determine what customers are over satisfied or visited at inappropriate times, the customers in the new route are given preference for staying. The resultant solution contains the additional information of the crossover operation without the redundancy or operation errors.

For this investigation the OB library is selected. The OB library contains very powerful and well constructed tools for the creation of evolutionary algorithms. It is written in C++ allowing for easier integration with existing simulation uses, all of which are written in C++, and the library allows the use of the vector data structure. The library is written in very strict object oriented protocol, meaning little work is required on the part of the user to get program specific details integrated into the overall programm structure, assuming they are written to the same OO standard. The selection of this infrastructure drives the code level requirement of all the program components as well as the data structures available.

4 EXPERIMENTAL DESIGN OBJECTIVES

The goal of any experiment is to contribute evidence to the hypothesis proposed. In this case, there are two hypotheses to test stemming from the objectives defined in Section 2; the proposed solution design for application to the VRPTW is valid across a spectrum of benchmark problems, and the solution design for application to the SRP produces valid results comparable to those obtained in the corresponding VRPTW benchmark. These objectives drive the experiment design such that a set of benchmarks are applied to the VRPTW and SRP solutions resulting in a set of valid solutions. These solutions then contain measurable metrics of total path length, total vehicle count, total wait time, and average path length (these metrics apply to both the VRPTW and SRP). In the case of the SRP the benchmarks are modified such that customer demand is an indication of vehicle count and not capacity demand, as in the VRPTW. Comparison of these metrics of performance allows for an intelligent comparison of the solution process to benchmark problems. Also, EA settings or parameters are determined based upon empirical experiments.

Three different algorithm designs, each with two options for selection strategies, are used in the experimental procedures. These three designs are NSGA2, SPEA2, and a biased elitism algorithm. The biased elitism algorithm uses no strategy to rank solutions instead using an elitist ordering procedure that is biased toward path length. The top number of individuals, equal to the population size, are selected from the population after genetic alteration. Each of these designs is then paired with either a random or tournament selection process. Recall that selection refers to how solutions are selected for genetic mutation. Tournament selection means some number of random individuals is selected from the population, with replacement, and ranked (biased by path length) with the top rank selected for alteration. The SRP experiments employ only the use of the tournament selection method; random selection is deemed more harmful to the SRP solution process from the fact that the genetic operators employ no local search techniques.

5 VRPTW and SRP Experiments

The most commonly used benchmarks for the VRPTW are the Solomon problems developed in 1987 (Solomon 1987). They exist in three different varieties; a random distribution of customers (R), clustered sets of customers (C), and hybrid (RC). Each of these three problems comes in dimensions of twenty five and fifty customers. In order to examine the effectiveness of the software as well as the impact of the multi-objective design, two problems from each type are tested, listed in Table 1. The use of this variety of problems illustrates the impact of problem type on the solution design as well as solution performance in different instances. The number designation of each problem constitutes the time windows that exist for that problem. Problems that begin with a one, such as R109, have small time windows, while R206 has much larger time windows.

Table 1: Solomon test problem selections

	Random		Cluster		Hybrid	
25 Tar	R206	**R109**	**C103**	C205	**RC107**	RC202
50 Tar	R206	R109	C103	C205	RC107	RC202

Each SRP test problem contains a set of target coordinates, target time windows, and vehicle capacity. The Euclidean distance between targets is considered to be the edge cost. The same problem selections are applied to the SRP solution modified in the demand column to ensure that each problem contains a realistic UAV requirement.

Algorithm effectiveness varies greatly as different parameters within the program are tuned. The settings for each algorithm type are determined from empirical analysis and literature review (Ombuki, Ross, and Hanshar 2006). The operator percentage indicates the chance that operator is used on an individual during the alteration phase. The more effective operators are used more often while the random operators are used less. The options for the algorithm used to solve the VRPTW problems are quit similar to those the option for the SRP algorithm which are listed in Table 2.

The SRP software experiments use the NSGA2 and biased elitism algorithms. The reason for this is that results from the VRPTW reveal a consistent dominance of these two methods over SPEA2. Each algorithm/problem experiment is run thirty times in order to ensure reliable statistical analysis. Each replacement strategy uses a tournament selection method. The population size and operator application percentages are different from the VRPTW settings in order to counter the SRPs fragile structure. More simple operations are performed to take the place of a few intelligent operations. Experiments are run against a small subset of the problems applied to the VRPTW (those entries bolded in Table 2).

Table 2: SRP GA Settings.

Operator	**Setting**
Random Crossover	50%
Split Mutation	25%
Vertical Swap Mutation	5%
Generation Limit	5000
Population Size	100
$\frac{\mu}{\lambda}$ ratio	2

6 VRPTW RESULTS AND ANALYSIS

VRPTW optimization occurs across three dimensions of total path length, total wait time, and number of vehicles used. Previous analysis, and the classical view, of this problem attempts to optimize path length and the number of vehicles used (Ombuki, Ross, and Hanshar 2006) or path length alone (Toth and Vigo 2001). Experiments performed in this investigation do not yield a single solution optimized in any one direction but rather a Pareto front of non-dominated values. In order to compare the results found here to those in the literature they are first shown in terms of the best path length found overall. The following box plots show the best path lengths available at the time of this writing compared to a distribution of values found from experimental trials (30 trials). Each box plot shows results for a single problem across six algorithm settings: SPEA2, NSGA2, Biased Single Objective; each of which uses either random or tournament selection. The wording used to express each of these settings is shown in Table 3. Each plot also shows the best answer for path length optimization found in Toth (Toth and Vigo 2001) and Diaz (Diaz 2007).

Where appropriate, a Kruskal-Wallis test using an alpha of 0.05 is used to further analyze performance for the different algorithms on specific problems. Following these box plots, solution space plots are shown across dimensions of path length and wait time in order to better examine algorithmic performance. The drive for this is that observing only path length can be misleading when examining MOEA performance. Each type of problem is defined in the experimental design discussion: random, cluster, and hybrid.

Table 3: Box plot label explanations for VRPTW experiments.

Plot Definition	Meaning
SPEA2 Tourn	SPEA2 replacement strategy using tournament selection for genetic operator application
NSGA2 Tourn	NSGA2 replacement strategy using tournament selection for genetic operator application
Bias SingleT	Biased Single Objective replacement strategy using tournament selection for genetic operator application
SPEA2 Rand	SPEA2 replacement strategy using random selection for genetic operator application
NSGA2 Rand	NSGA2 replacement strategy using random selection for genetic operator application
Bias SingleR	Biased Single Objective replacement strategy using random selection for genetic operator application

Random Distribution Problem: The difference in performance between high and low dimension problems is considerably different. Figure 6 indicates the results for 50 customers where NSGA2 is observed to return results closer to the best answer, followed by the biased single objective algorithm, with SPEA2 doing worst. A Kruskal-Wallis statistical analysis performed in Matlab confirmed these visual observations.

The performance observation per algorithm is repeated in the R206 problem. NSGA2 again manages to pull ahead in terms of the path length objective and along with the biased algorithm approaches the best solution in the higher dimension 50 customer problem. The consistent convergence of solutions for NSGA2 using tournament selection is observed.

Cluster Distribution Problem: Within the cluster benchmarks the path length objective becomes less consistent in returns. All methods are converging close to the best answer with NSGA2 actually achieving it in a few trials. Increasing the dimension of the problem, in Figure 7, causes a return to the performance seen so far, with no algorithm approaching the best solution.

For the lower dimensional problem, convergence is achieved using the biased algorithm with a wide dispersion of points using SPEA2 or NSGA2. over higher dimensional problems the results are quite similar to that of Figure 7 which reflects almost the same performance. NSGA2 returns better statistical results. It is interesting to note the results of NSGA2 with random selection returning with such consistent results. This can be most likely attributed to the nature of the cluster problems working well with the genetic operators used.

Hybrid Distribution Problem: The hybrid problem would seem to represent the most difficult landscape to work in. However for the lower dimensional problem, no algorithm had particular trouble arriving at the optimal solution. This is less true for the fifty dimension problem in Figure 8 as seen from the results being further from the optimal value line. NSGA2 and the biased algorithm returning consistent values with tournament selection being the deciding factor in superior performance. Further generational development would most likely force the solution closer to the optimal. NSGA2 achieves statistically better results by a small margin, further leading to the conclusion of its usefulness in developing solutions. However, previous results also show consistent returns using the biased algorithm, meaning no one strategy dominates overall.

The hybrid problem solution for 25 customers again indicates the convergence of the biased algorithm occurs while NSGA2 and SPEA2 maintain a larger coverage. This is seen in the disconnected Pareto fronts of Figure 9. Statistical comparison plots indicate consistent results across NSGA2 and the biased algorithm. SPEA2 is again beaten in this particular performance measure due to is dominance selection criteria.

Results from only observing the path length objective can be informative but also slightly misleading. It might be assumed

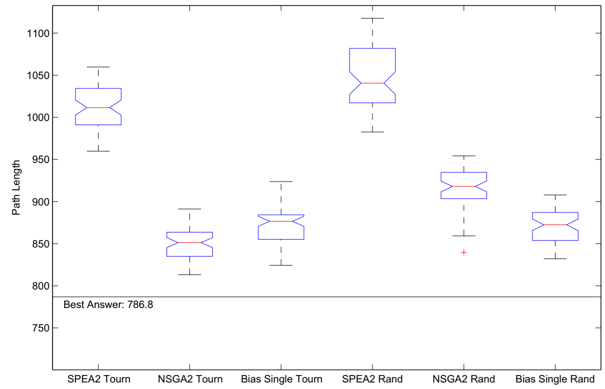

Figure 6: Trial results for random distribution problem R109 with 50 customers

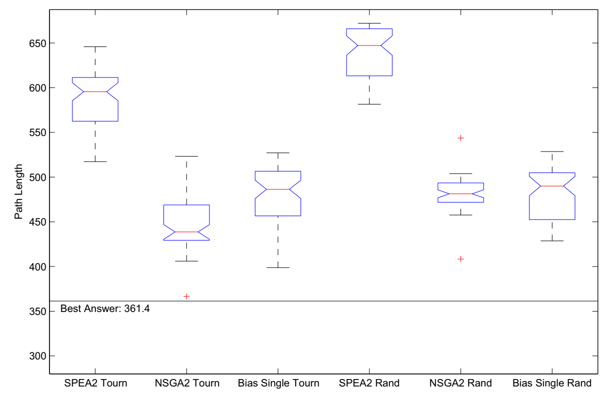

Figure 7: Trial results for cluster distribution problem C103 with 50 customers

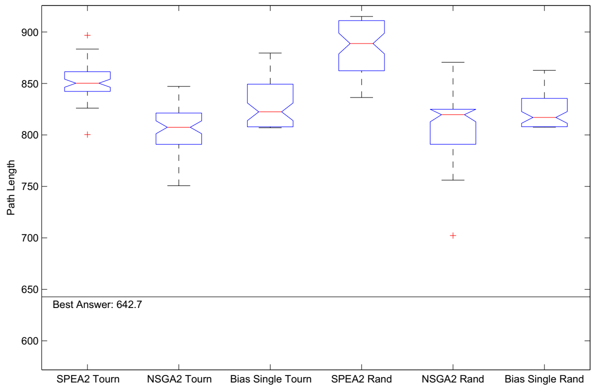

Figure 8: Trial results for hybrid distribution problem RC107 with 50 customers

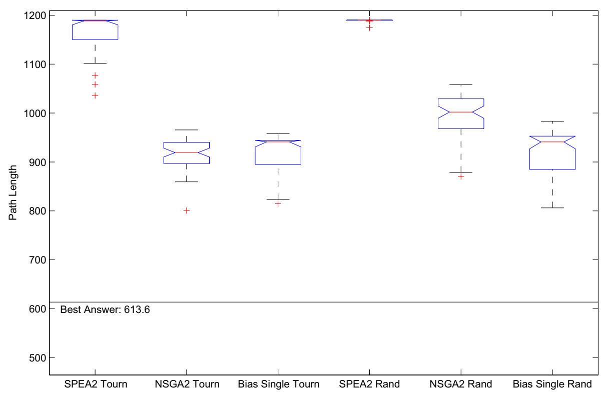

Figure 9: Trial results for hybrid distribution problem RC202 with 50 customers

that SPEA2 is being outperformed in all problem instances, and in terms of path length it is. Analysis of the non dominated approximated Pareto front generated by the NSGA2 and SPEA2 trials show a return of results consistent with what one would expect from a VRPTW multi-objective problem.

The conclusion to be made is that the multi-objective solution is effective in returning a broad range of results and that these results are pushing the front of the problem. It is therefore not odd that the multi-objective approach did not return a near optimal value for path length. The returned value represents the solution space for the objectives selected. The fact that the returned values are close (i.e within 10 percent in most cases) to the highest benchmark value, shows the validity of the MOEA approach as being able to find the optimal value for a single objective, while also optimizing across the range of objectives. In short, it appears that multi-objective optimization is appropriate for this particular routing problem.

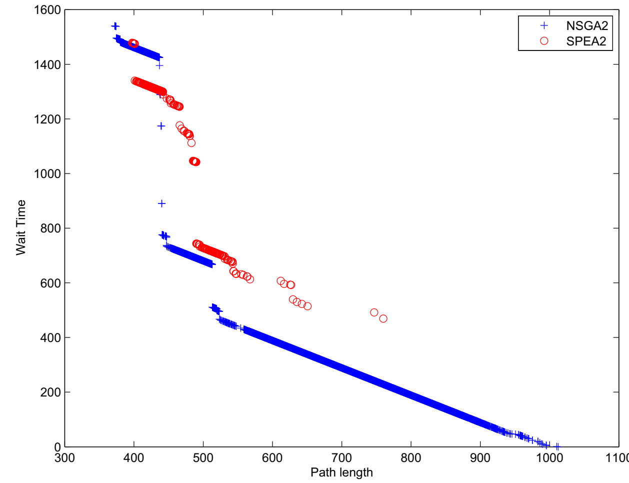

Figure 10: Non-dominated front comparing NSGA2 and SPEA2 for RC205 with 25 customers

7 SRP RESULTS AND ANALYSIS

SRP optimization occurs across dimensions of total path length, wait time, the number of vehicles used, and average path length. As this problem formulation is unique to this investigation there are no readily comparable results. Though the problems differ in formulation the objectives remain the same between the SRP and VRPTW. As such, the results obtained from the VRPTW solution are compared in order to illuminate the hypothesis that the SRP represents a superior problem model in terms of individual vehicle operation and mission optimization. As in the previous section results are organized in box plots representing trial results for the selected problem. Definitions for the labels used in the plots are the same as for the VRPTW. A comparison of the best results from the VRPTW solution are also shown both in terms of total path length and average path length (even though average path length is not an optimized objective in the VRPTW).

Random Distribution Problem: The total and average path length returned by NSGA2 and biased algorithm indicates that The biased algorithm converges while NSGA2 retains a larger spread of the solution space. Neither NSGA2 or the biased algorithm perform significantly better than one another.

For the cluster distribution problem, The return of total and average path length between NSGA2 and the biased algorithm with further comparison between the same VRPTW indicates a decreasing ability to handle this particular type of problem. This behavior should be expected as the VRPTW solution contains heuristic operators that deal specifically with clustered targets, while the SRP does not. The equal performance of the biased algorithm and NSGA2 for the VRPTW is obtained. Even with the high constraints of the SRP problem model it is still possible to return per vehicle path lengths of the same distance.

8 CONCLUSIONS

Results from the VRPTW experimentation showed a consistent return of results across a broad spectrum of problems. Optimization along the path length objective showed less than optimal results, however when combined with a view of the achieved non-dominated front, it is clear that the MOEA strategy is working correctly. Further comparison between results shows the NSGA2 algorithm performing better or as well as the biased algorithm. However, in some cases the biased solution converged early while the MOEA approaches maintained a breadth of search in the solution space. These results lead to the conclusion that the MOEA solution method developed and implemented here is effective at optimization over a range of different problems.

Even with a multi-objective design, optimization of the path length objective still approaches optimal value. NSGA2 is able to achieve a path length value within ten percent of the optimal value, and is even closer in some cases. It can be concluded that even while the algorithm is optimizing across multiple objectives the returns for a single objective are no being compromised, as evidenced by NSGA2s performance on the various benchmark problems (Pohl 2008).

With the validation of the MOEA design in place, obtained through analysis of results over VRPTW benchmark problems, attention can then be turned to to the SRP problem model. Results for the SRP again show consistent returns of total and average path length. Average path length is then compared to the average path length returns for the VRPTW solution in order to show that the SRP achieves comparable results, which it does. That the average path length returns for the SRP are comparable indicates the merit of the model as a per vehicle optimization strategy. The purpose of the SRP as model is to develop time constrained routes between many different targets each of which requires some number of vehicle visitations. It is no sunrise that total path length is greater for the SRP returns, it would have to be, what is important is that the returned solution does not require any one vehicle to visit a large number of points, as would be the case in the VRPTW.

The benchmarks used in these experiments should also be considered reflective of real world problems and not merely contrived problems. A real world mission for a compliment of UAVs can conceivably contain 20 or more targets, to which these benchmarks affectively match. The SRP model shows capability not only as a combinatorics formulation but also as an applicable model for real world problem formulation, as the solutions shown here validate a capability to return consistent solutions.

Acknowledgments

This effort is in support of the AFIT Advanced Navigation Technology (Ant) Center (Director: Dr. John Raquet). The sponsors of this research are the Air Force Research Laboratory (AFRL) Information Directorate (Dr. Robert Ewing) and the Sensors Directorate - Virtual Computing Laboratory (Mike Foster).

REFERENCES

Coello Coello, C. A., and G. B. Lamont. (Eds.) 2004. *Application of multi-objective evolutionary algorithms*, Volume 1 of *Advances in Natural Computation*. World Scientific Publishing.

Coello Coello, Carlos A.; Lamont, G. B. V. V. D. A. 2007. *Evolutionary algorithms for solving multi-objective problems*. 2 ed. Genetic and Evolutionary Computation. Springer.

Corner, J. 2004. Swarming reconnaissance using unmanned aerial vehicles in a parallel discrete event simulation. Master's thesis, Air Force Institute of Technology.

Diaz, B. D. 2007. The VRP Web: Solution techniques for VRP. `http://neo.lcc.uma.es/radi-aeb/WebVRP/m`.

Lou, S.-Z., and Z.-K. Shi. 2006, Aug.. A new method for multi-depot vehicle routing problem with time windows. In *Machine Learning and Cybernetics, 2006 International Conference on*, 2503–2509.

Ombuki, B., B. J. Ross, and F. Hanshar. 2006. Multi-objective genetic algorithms for vehicle routing problem with time windows. *Applied Intelligence* 24 (1): 17–30.

Pohl, A. J. 2008. Multi-objective UAV mission planning using evolutionary computation. Master's thesis, Air Force Institute of Technology.

Russell, M. A., and G. B. Lamont. 2005. A genetic algorithm for unmanned aerial vehicle routing. In *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, 1523–1530. New York, NY, USA: ACM Press.

Slear, J. N. 2006. AFIT UAV swarm mission planning and simulation system. Master's thesis, Air Force Institute of Technology.

Solomon, M. M. 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Oper. Res.* 35 (2): 254–265.

Tavares, J., P. Machado, F. B. Pereira, and E. Costa. 2003. On the influence of GVR in vehicle routing. In *SAC '03: Proceedings of the 2003 ACM symposium on Applied computing*, 753–758. New York, NY, USA: ACM Press.

Toth, P., and D. Vigo. (Eds.) 2001. *The vehicle routing problem*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics.

AUTHOR BIOGRAPHIES

ADAM J. POHL is a 2nd Lt. in the USAF. He graduated in March 2008 from the Air Force Institute of Technology, WPAFB, Dayton, OH, with a Master of Science in Computer Engineering.

GARY B. LAMONT is a Professor in the Department of Electrical and Computer Engineering, Graduate School of Engineering and Management, Air Force Institute of Technology; B. of Physics, 1961; M.S. in Electrical Engineering, 1967; Ph.D., 1970; University of Minnesota. Dr. Lamont's research interests in computer science and engineering include heuristic search algorithms, bio-inspired search methods, and parallel and distributed processing. Tel: 937-255-3636 x4718; Email: gary.lamont@afit.edu