Applying Evolutionary Multi-Objective Optimization to Mission Planning for Time-Sensitive Targets

Brad Rosenberg
Charles River Analytics, Inc.
625 Mt. Auburn Street
Cambridge, MA 02138
brosenberg@cra.com

Janet Burge
Charles River Analytics, Inc.
625 Mt. Auburn Street
Cambridge, MA 02138
jburge@cra.com

Paul G. Gonsalves
Charles River Analytics, Inc.
625 Mt. Auburn Street
Cambridge, MA 02138
pgonsalves@cra.com

ABSTRACT

This paper describes an approach to air campaign mission planning using evolutionary multi-objective optimization. With the landscape of warfare constantly changing, timely and accurate employment of air assets for military operations has become even more crucial. Of particular importance is in addressing time-sensitive and time-critical targets. Such operations require a rapid search of potential mission plans to evaluate their ability on an array of objectives. This type of system design problem, consisting of a large solution space and complicated fitness landscape, has proved to be approached successfully using evolutionary algorithms. Additionally, the presence of potentially multiple conflicting objectives lends to the suitability of using multi-objective optimization techniques. This paper describes our preliminary experiments using both aggregation and dominance-based approaches to evolutionary multi-objective optimization for addressing time-sensitive/critical targets.

Categories and Subject Descriptors

G.1.6 – Optimization; I.2.8 – Problem Solving, Control Methods and Search; I.6.3 – Simulation and Modeling Applications

General Terms

Algorithms, Design

Keywords

Genetic Algorithms, Evolutionary Multi-Objective Optimization, Mission Planning, Time-Sensitive Targeting.

1. INTRODUCTION

The landscape of warfare is constantly changing. Where past engagements have normally focused on out-attriting enemy forces, current and future operations have been, and most likely will be, characterized by more timely and accurate employment of air assets. The recent increase in the capabilities of precision guided munitions and the further development of strike warfare platforms and tactics have put this focus on addressing time-sensitive and time-critical targeting. Time-sensitive targets (TSTs) are targets where modifiers such as, "emerging, perishable, high-payoff, short dwell, or highly mobile" can be used [19]. Time-critical targets (TCTs) further the criticality of TSTs with respect to achievement of mission objectives and a limited window of opportunity for attack. One such example of this type would be receiving intelligence about a clandestine meeting where leaders of a terrorist organization would gather.

To fully realize the benefits of these emerging technologies and address the challenges of time-sensitive targeting, decision-support systems are needed to assist warfighters in this effort. We are currently developing a system to support this growing need. The focal point of our approach is centered around using evolutionary multi-objective optimization (EMOO) to evolve mission plans for air operations regarding TST/TCTs. Described in this paper is our initial approach into this area [8][9], the results of our preliminary experiments, and a discussion of our future plans. The paper is organized as follows. Section 2 describes why we chose evolutionary multi-objective optimization to approach this problem. Section 3 outlines the component architecture of our system. Section 4 describes the representations used and evolutionary techniques employed in our preliminary experiments. Section 5 concludes the paper with relevant conclusions and future work.

2. APPROACH

Our approach focuses on using evolutionary multi-objective optimization to evolve mission plans to address the needs of air operations regarding TST/TCTs. Using evolutionary algorithms for engineering problems, such as air operation mission planning, has been met with considerable success in the past. In general, systems design problems that require optimizing a function depending on a large number of parameters pose a significant challenge. Large solution spaces and complicated fitness landscapes are difficult for standard gradient-based approaches. Evolutionary algorithm-based approaches, however, do not primarily rely on gradient information and can lend toward a more successful methodology.

Optimization of air campaign mission plans, like many real world problems, is characterized by multiple potential conflicting objectives. As an example, the shortest time route to the strike targets may impose undue vulnerability on the strike package, or the best weapons loadout may inversely effect the radar cross section of strike aircraft. Over the past 15 years or so a large volume of research within the optimization community has focused on the multi-objective optimization problem [2][4]. Two specific approaches have normally been utilized, local search methods [3][6] and evolutionary algorithms. Primarily, evolutionary algorithms have outperformed local search methods in both speed and quality of solutions [14].

Several methods are available to handle multi-objective optimization using evolutionary techniques. Coello, et al [2] classifies these into first and second-generation methods. First generation methods include aggregation, a simple approach where all the objectives are combined into a scalar value (e.g. weighted sum). While being simple and easy to implement, this approach can fail to generate optimal solutions for given search spaces. Schaffer [18] proposes the use of sub-populations that optimize each objective separately. The approach was successfully used by Rogers [16] to design the optimum placement of aircraft actuators, but in general does not guarantee Pareto optimum solutions. Goldberg [7] extended Schaffer's work to support the generation of Pareto optimum solutions via the incorporation of a ranking procedure and a mechanism to maintain diversity in the population. Second generation techniques have focused on enhancing the efficiency of evolutionary algorithms. For example, Knowles and Corne's Pareto Archived Evolution Strategy (PAES) [13] combines a local search method for the generation of new candidate solutions while also utilizing population information in its selection procedure. Inefficiencies in multi-objective optimization algorithms are addressed by Jensen [12] who proposed a new algorithm for non-denominated sorting, the method used for fitness assignment in many multi-objective evolutionary algorithms.

Based on the aspects of the problem definition, the success of these approaches by others to solve similar problems, and stemming from our own past achievements with using genetic algorithms in the military domain [10][11][15][17], it seemed suitable to utilize evolutionary multi-objective optimization for the issue of rapid air operation mission planning for TST/TCTs.

3. COMPONENT ARCHITECTURE

In order to support mission plan generation through evolutionary multi-objective optimization for rapid air operations regarding TST/TCTs, the following component architecture was developed (See Figure 1). This architecture consists of four major components: A component to specify the given scenario, a genetic algorithm-based optimization component to evolve mission plans, an abstract wargamer to evaluate mission plans generated during the evolutionary process, and finally a user interface to support user interactions with the system.

3.1 Scenario Specification

The Scenario Specification component is responsible for gathering incoming information from various sources to formalize the problem the system is attempting to solve. These sources can include information gathered during the intelligence preparation of the battlefield (IPB) process, plan related information from command and control (C2) systems within an Air Operations Center (AOC), and other related sources. This information includes a description of the physical battlespace environment including both terrain and airspace management features; current battlespace state including order of battle or available friendly assets, likely enemy courses of action (COAs) that may pertain to the particular strike mission and any other relevant battlespace

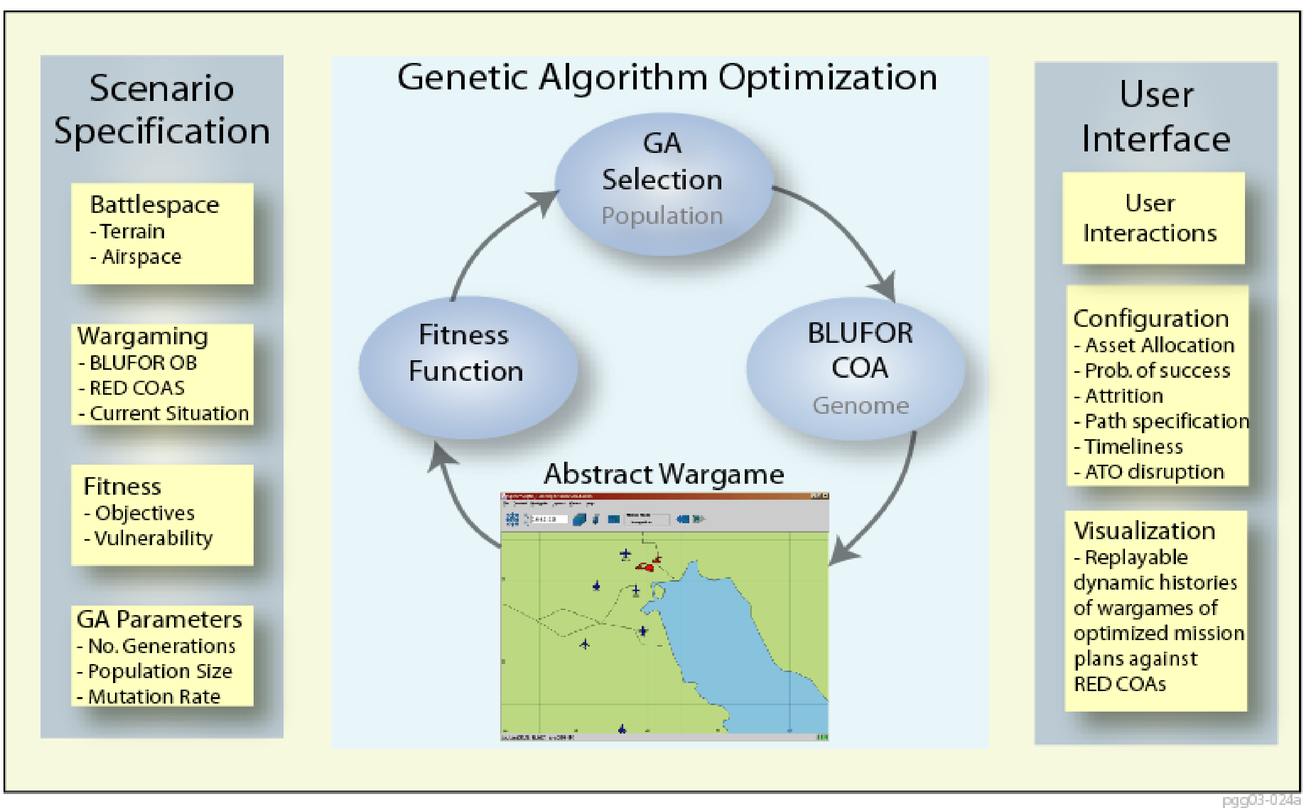

Figure 1: Component Architecture

states; and commander's intent information to support fitness specification (e.g. time constraints for destroying the target, prioritization of targets, allowable level of vulnerability, etc.). In our current implementation, this component also supplies the air tasking order (ATO) to describe the current state of occupied friendly assets for considering disruption to other missions.

3.2 Genetic Algorithm-based Optimization

The Genetic Algorithm-based Optimization component is the key component that drives the mission planning generation process supplied by the developed system. In this component, the problem supplied by the scenario specification is solved using evolutionary multi-objective optimization techniques. Mission plans encoded as gene strings are initially generated and then managed by an evolutionary algorithm. For evaluation, each mission plan is sent to the Abstract Wargamer to be played out in a simulated environment. Results from this simulation are used to calculate fitness on various objectives and used during the selection, recombination, and mutation process. When the optimization process is complete, selected solutions from the final population are returned to the user interface for playback and human-based evaluation.

3.3 Abstract Wargamer

The Abstract Wargamer component is responsible for playing out or "wargaming" each generated solution within the population at each iteration of the evolutionary algorithm. The key here is that the Abstract Wargamer only encompasses those key features that define the particulars of a mission plan at a minimum fidelity level to ensure that the plans are representative and useful as a decision support tool to address time-sensitive targeting. Furthermore, this is done to support a quick turnaround time on the optimization process, an essential requirement of the system we are developing.

The simulation provided by the Abstract Wargamer utilizes the Open Experimental Platform (OEP) developed by Boeing. The OEP is based on the Boeing C4ISim, a simulator that models the collection, processing, and dissemination of battlefield information. The OEP includes a enemy controller to model enemy air defenses and behaviors. Simulation is performed via a Monte Carlo method, stochastically determining the operational success of both friendly and enemy weapon systems. Visualization of the simulations provided by the OEP utilizes BBN's OpenMap technology, An example of the visualization within our system is illustrated in Figure 2.

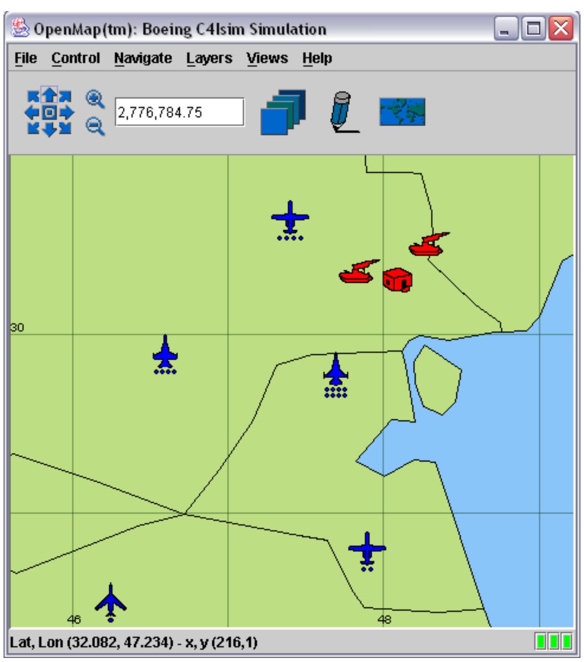

Figure 2: OpenMap Visualization of Simulation

3.4 User Interface

The User Interface component is responsible for allowing interactions with the user to help determine the scenario specification, provide parameters for the optimization process, and visualize the generated mission plans returned at the completion of the evolutionary algorithm. The user of the system can then provide a human-based assessment of selected plans given the internal simulator. Additionally, the user will be able to send the generated mission plan to a higher-fidelity external simulator for further evaluation.

4. EVOLUTIONARY PROCESS

We shall now describe the methods used to generate mission plans for the described system through evolutionary multi-objective optimization. First, we shall describe the manner in which we create an abstract battlespace representation. We will then describe the representations developed for mission plans. Finally, we will describe the various evolutionary techniques employed to evolve mission plans.

4.1 Abstract Battlespace Representation

A crucial need exists to restrict the potential solution space from an infinite continuous range to a discrete range that can be adequately explored by an evolutionary algorithm. To accomplish this, in our initial experiments we superimposed a square coordinate grid on the portion of the battlespace around the target that encompasses both the target and the range of effectiveness for the enemy assets defending it. Figure 3 shows an example of a grid, with paths from asset starting positions to the target. In our actual implementation, a 10x10 grid was chosen.

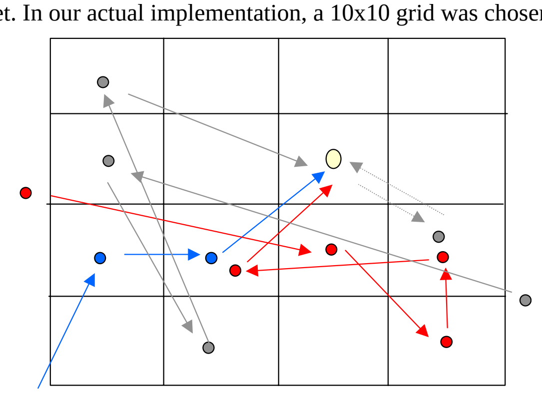

Figure 3: Abstract Battlespace Grid

4.2 Plan Representation

4.2.1 Initial Representation

In our initial experiments, a minimum representation was used for our encoding. Each aircraft group was given a field indicating if it was “active” (i.e. included in the plan) by a simple binary flag. Additionally, each aircraft group was also given a fixed number of grid coordinates indicating two-dimensional waypoints between its initial position and the target destination. If a path intercepts the cell of the target, the path terminates immediately in that cell. In our initial experiments, the fixed number of waypoints was set to four to provide a rich, yet feasible, solution space to explore. Each aircraft group was given a static location within the chromosome to avoid the need for additional information. An illustration of the representation for a single aircraft group is supplied in Figure 4.

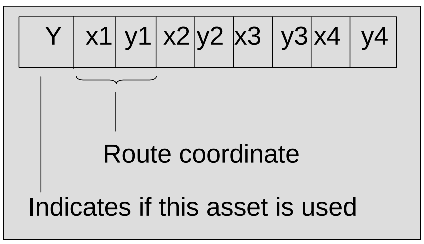

Figure 4: Initial Plan Representation

4.2.2 Modifications and Improvements

As our initial experiments completed, several questions arose as how to improve the representation to better characterize potential solutions as well as ease the evolutionary process in finding fit solutions. One question posed what influence the proximity of the active flag to the path for each aircraft group had. In response, an experiment was conducted in which the active flag fields for each aircraft group were collected together and placed at the beginning of the chromosome. Each aircraft group’s path was then set in order following the active flag fields in the same manner as in the initial representation. Given the same evolutionary technique, each run performed comparatively to the prior representation. This was a puzzling result given that the hypothesis suggested the experiment would result in either an improvement or a decline in the findings. The central speculation, given the intricacies of the domain, was that the active flag fields influence the fitness of each objective to a higher degree than the path of the aircraft group. We are currently analyzing these results to determine why proximity does not seem to have a profound effect on the results. It is possible that this representation improved one property while worsening another, such as improved asset allocation but worsened path formation.

We are also currently investigating developing a more intricate gene representation to provide a more realistic picture to properly correlate to our final domain. Current efforts are focusing on including three-dimensional waypoints to include altitude, including fuel levels and other logistics, and considering inter-asset synchronization requirements to support a more collaborative approach to the mission plan. Additionally, we are exploring replacing explicit path formation via waypoints with a strategy of approach that will be translated into a path during simulation.

4.3 Evolutionary Techniques

4.3.1 Plan Evaluation

Each generated mission plan from the evolutionary algorithm is played out within the Abstract Wargamer to simulate the effects of that mission plan in the environment. In our current implementation, the following objectives are retrieved from data supplied by the Abstract Wargamer and calculated:

- Probability of destroying the target
- Vulnerability of friendly assets
- Time required to accomplish the mission
- Disruption to the Air Tasking Order

Each objective is calculated and normalized to a real number between 0.0 and 1.0. For simplification, each objective is set to be maximized through inversion where need be. Each objective will be referred to in the following manner: *Success* (*S*), *Vulnerability* (*V*), *Time* (*T*), and *Disruption* (*D*).

Success is defined as the probability that a given mission plan will successfully reach and destroy its target. As described above, the mission plan is simulated using the OEP simulation capabilities. Due to the stochastic nature of the Monte Carlo simulation, several runs are completed and the results averaged together.

Vulnerability is defined by two separate means. The initial vulnerability metric is a straight calculation of the number of friendly aircraft destroyed during the simulation. Again, several runs are completed and results averaged together to receive an average number of friendly aircraft destroyed during mission execution. An additional vulnerability metric used is by integrating the vulnerability described by the path of each friendly aircraft. This computation is performed by comparing the inherent vulnerability of the friendly aircraft to the vulnerability characterized by the threat at each location in the abstract battlespace grid. Friendly aircraft are initially given associated discrete values of vulnerability from a lookup table based on the aircraft type. The threat at each location is then given by the capabilities of the enemy aircraft and enemy artillery compared to the inherent vulnerability of the friendly asset to compute the vulnerability. Figure 5 illustrates an example of the varying threat levels in the abstract battlespace grid for a sample friendly aircraft.

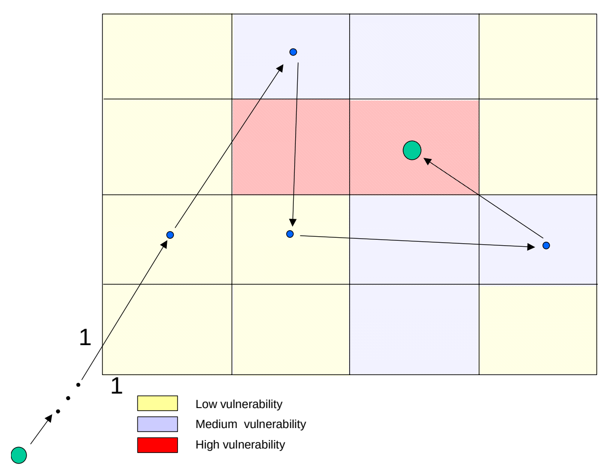

Figure 5: Threat Levels in Abstract Battlespace Grid

Time is defined by how long the given mission plan takes to complete. This is calculated by how long it takes from the beginning of the mission plan until the last aircraft included in the plan reaches its target. This calculation is then inverted and normalized by comparing it to the maximum possible duration of a mission plan. Missions that take less time are given higher levels of fitness for this objective.

Disruption is defined by the number of friendly aircraft utilized in the mission and the associated priority levels of the previous targets of those aircraft governed by the retrieved ATO. Plans that use fewer aircraft and with lower priority for their original target are rated higher than those who divert assets from higher priority targets.

In addition to the objectives outlined above, we are currently performing experiments to include additional objectives as well as further detailing current objectives. One element we wish to bring into the picture is the concept of secondary effects due to the weaponry used and the location of targets. An additional desirable objective would be the minimization of cost and weapon expenditure. Furthermore, the degree to which the given mission plan violates airspace restrictions is an important attribute. Additionally, there are other performance metrics to determine the probability of success under consideration. For example, the performance data of an asset is useful to integrate into the probability of success computation. The weapon load effectiveness is an additional piece of data we plan on integrating. Also in consideration is the proximity to the target window that a given mission plan executes within.

4.3.2 Aggregation-based Multi-objective Optimization

In our initial experiments, we began by using a simple aggregation-based method of multi-objective optimization. Here, fitness was calculated by weighting (w_i) each objective and summing up all objectives. The applied weights were supplied by the user of the system to designate the priority of each objective in relation to the others. Equation 1 describes the simple mathematical formulation for this aggregation-based fitness.

This method was then tested using a scenario developed by a domain expert. In this scenario, a single, fixed target was used surrounded by two surface-to-air missiles (SAMs). This form of the scenario was developed to serve as a test problem by which to examine the capabilities of the evolutionary technique given. Table 1 shows the status of each friendly asset to be potentially included in the mission plan.

Equation 1: Aggregation-based Fitness

$$Fitness = w_S \cdot S + w_V \cdot V + w_T \cdot T + w_D \cdot D$$
$$\sum w_i = 1$$

Several experiments were run using a standard genetic algorithm with elitism, roulette wheel selection, N-point crossover with N=2 and crossover rate of 0.7, a standard jump mutation operator with a 0.05 mutation rate, a population size of 25, each run given 20 generations, and 10 simulations. Each set of experiments used different combinations of weightings for the objectives given by the problem. A selection of the fitness results from several runs is summarized in Table 2.

Table 1: Friendly Asset Status

Asset	Allocation	Status
2 x 4 x F-16C	Priority 0	Coming off target
2 x 2 x F-15E	Priority 2	Taxing, 250 nm from target
1 x 4 x A-10	Priority 2	80 nm from target
1 x 1 x B1-B	Priority 1	150 nm from target
2 x 2 x F-18G	Priority 3	80 nm from target
1 x 2 x A-10	Priority 0	On strip alert, 150 nm from target

When a high weight was given to *Success* (*S*), the probability of destruction given was almost always definite. However, it should be noted that the probability of approaching the target and being successful in its destruction is inherently easy given the parameters set forth by the simulation. Even when the priority was not given to Success, the destruction of the target was fairly certain arriving at a probability around 1. When a high weight was given to *Vulnerability* (*V*), the average number of friendly aircraft destroyed varied between runs. During some instances, the average number destroyed fell much lower than when priority was given to other objectives. In other instances, however, the average number of friendly aircraft destroyed was about equal to when higher priority was given to other objectives. When a high weight was given to *Time* (*T*), the average time required dropped

Table 2: Selection of Aggregation-based Results Given Various Priority Ratings

Priority	Asset Groups	Fitness	Probability of Destruction	Average Assets Destroyed	Average Time Required
Success	F/A-18G-1, F-16C-1	0.861	1	0.4	0.504
	F/A-18G-1, F-16C-2	0.852	1	0.5	0.422
Vulnerability	F-16C-2, F-16C-1	0.849	1	0.5	0.431
	F/A-18G-1, F/A-16C-2	0.828	1	0.2	0.476
Time	F-16C-1	0.865	0.99	0.4	0.112
	F-16C-1	0.845	0.99	0.5	0.114
Disruption	F-16C-2	0.985	1	0	0.211
	F-16C-2	0.981	1	0	0.268

considerably and the algorithm was able to minimize the number of friendly aircraft used and find a short path to the target. Coincidently, the aircraft selected also minimize the disruption to the ATO by utilizing aircraft with low priorities specified by the order of battle. When a high weight was given to *Disruption* (*D*), low priority aircraft were utilized. Additionally, these runs seemed to result in a quite accurate path formulation as the average number of assets destroyed was minimized to 0. The alternative vulnerability calculation, however, remained within the normal bounds.

Based on these results, several findings were realized. First, there are some clear interactions between the differing objectives that are not always contradictory given this scenario. Primarily, there seemed to exist a relationship between both time and disruption to push evolution to select a minimum number of aircraft and to utilize those close to the target and with low priority ratings. Further study is underway in this area by staging more deliberate scenarios and reexamining the underlying objective calculations. An additional finding was that when a priority was placed on vulnerability, this evolutionary technique failed to consistently provide a better vulnerability rating. We speculate that for this objective, the fitness landscape may be more complex, causing the optimization procedure to culminate at local maxima. Finally, and perhaps most importantly, further runs resulted in the identification that the final population often contained multiple mutants of single individuals. This clustering behavior suggests that this evolutionary strategy consistently results in premature convergence. Since the goal of the optimization process is to provide an optimal, yet diverse, population of solutions for the end user, a crucial need exists to implement a method of diversity preservation.

4.3.3 Dominance-based Multi-objective Optimization

Based on the preliminary tests, the aggregation-based multi-objective optimization approach as described in the previous section did not adequately explore the search space, resulting in non-optimal solutions on several runs. Additionally, the process consistently fell victim to premature convergence. In order to maintain a diverse set of solutions, we looked towards other evolutionary strategies. A decision was made to move from an aggregation-based approach to a dominance-based approach. In particular, we chose to implement the NSGA-II [5] algorithm because of its popular use, speed, ease of implementation, and ability to maintain a diverse set of solutions through non-dominated sorting and crowded distance estimation.

NSGA-II was developed by in response to criticisms of the original NSGA algorithm, one of the first multi-objective evolutionary algorithms (MOEAs) created. The revised form reduces computational complexity, includes a form of elitism, and removes the need of a sharing parameter. At each iteration of the algorithm, NSGA-II combines parent and child populations. The combined population is then sorted using a fast non-dominated sorting algorithm. During this process, each individual in the population is given a rank as to which Pareto front it belongs. Each Pareto front contains individuals which non-dominate each other. Pareto dominance is defined as an individual which is greater or equal than another individual on all objectives and greater on at least one objective. The individuals included in a Pareto front are removed from consideration when determining subsequent fronts. Once ranked by Pareto front, each front is subjected to a crowded distance calculation and sorting algorithm. This algorithm maintains diversity by ranking each individual by their proximity to other individuals on all objectives. Higher distance values represent more diverse solutions. Each front that can fit in the parent population is included. Individuals from the front that follows are then included in the parent population based on their crowding distance until the parent population is filled. Tournament selection (based on the crowded-comparison operator), crossover, and mutation operators are then used to generate the new child population.

We have included the NSGA-II algorithm in our system and performed some initial experiments to examine its performance compared to our previous aggregation-based approach. In our implementation, the same four objectives were used and each objective treated of equal importance as described by the NSGA-II process. Binary tournament selection based on the crowded-comparison operator, N-point crossover with N=2 as well as uniform crossover, and simple jump mutation were used. Runs processed for 50 generations, used population sizes of 50 individuals, and evaluated for 10 simulations. Initial tests have

Table 3: Single Run Results with NSGA-II

ID	# Assets	Asset Groups	Probability of Destruction	Average Assets Destroyed	Average Time Required
1	4	F-16C-1	1	0	0.091
2	4	F-16C-1	1	0.2	0.118
3	4	F-16C-2	1	0.8	0.100
4	6	F/A-18G-1, F-16C-2	1	0	0.300
5	6	F/A-18G-1, F-16C-2	1	0.4	0.286
6	2	F/A-18G-1	1	0	0.245
7	4	F-16C-1	0.24	0.2	0.102
8	4	F-16C-2	1	0.4	0.104
9	4	F-16C-1	1	0.6	0.102
10	4	F-16C-1	1	0.2	0.118

risen to a much better performance in terms of diversity. Selected runs have resulted in a spread of solutions that exhibit higher fitness levels in single objective domains than could be obtained with aggregation-based optimization. An example of the results from a single run is displayed in Table 3. The most dominant solution discovered by this algorithm was mission plan ID 1, which uses a low-priority asset group, has a high probability of success, loses no assets during simulation, and is completed in the least amount of time. Other solutions vary in tradeoffs among objectives. Mission plans ID4 and ID5 use higher-priority asset groups that result in less vulnerability but are higher average time required. Mission plan ID7 uses a low-priority asset group and has a low vulnerability rating and low time required, but has a low probability of success.

One of the downfalls, however, of the initial experiments utilizing the NSGA-II approach is that each objective is treated equally. As a result, it is not possible to provide the end user with a means by which to give preference to certain objectives and restrict the importance of other objectives. Recent work in the field has identified this limitation and is currently developing means to include user preferences in dominance-based multi-objective optimization. Of particular interest is the work discussed in [1], since it describes a guided evolutionary multi-objective evolutionary algorithm and a way to augment NSGA-II in using a biased crowding operator and the concept of guided dominance. Our future work includes utilizing this developed technique in the hopes that we can supply user preferences to the success of the dominance-based approach in this work.

5. CONCLUSION

We have outlined a system for the generation of air campaign mission plans using evolutionary multi-objective optimization and described the approaches and results of our preliminary experiments. Our aggregation-based approach was met with some success, but during many runs failed to locate the more optimal solutions. It was hypothesized that this approach did not fully explore the search space and was leading the evolutionary process to prematurely converge. To address this issue, we shifted to using a dominance-based approach, specifically the NSGA-II algorithm. Runs using this technique resulted in more optimal solutions, and also returned a more diverse collection for the end user. However, NSGA-II lacks a method by which the user can inject preferences into the evolutionary process, a largely required aspect of our system.

Future work in the development of this decision-support tool will focus on various fronts. First, we are looking towards experimenting with different representations for the encoding of our mission plans. Such representations might include a more abstract depiction of paths, encoding strategy instead of explicit route methods. Additionally, we are looking towards using more involved scenarios to test and validate the approach. Primarily, however, our focus will be on studying, implementing, and modifying existing multi-objective evolutionary algorithms and testing their success for our defined problem.

6. ACKNOWLEDGMENTS

The work performed in this paper was sponsored by the Air Force Research Laboratory (Contract Number: FA8750-04-C-0174).

7. REFERENCES

[1] Branke, J. and Deb, K., *Integrating User Preferences into Evolutionary Multi-Objective Optimization*. KanGAL Report No. 2004004, Indian Institute of Technology, Kanpur, India, 2004.

[2] Coello, C., Veldhuizen, D. V., et al., *Evolutionary Algorithms for Solving Multi-Objective Problems*. New York, Kluwer, 2002.

[3] Czyzak, P. and Jaszkiewicz, A., "Pareto simulated annealing - a metaheuristic technique for multiple objective combinatorial optimization," *Journal of Multi-Criteria Decision Analysis 3*, 1998, 34-47.

[4] Deb, K., *Multi-Objective Optimization using Evolutionary Algorithms*, New York, NY, Wiley, 2001.

[5] Deb, K., Agrawal, S., Pratap, A., Meyarivan, T. A Fast Elitist Non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. *Proceedings of the Parallel Problem Solving from Nature VI Conference*, 16-20 September, 2000. (Paris, France), 849-858.

[6] Gandibleux, X. , Mezdaoui, N., et al., A Tabu Search Procedure to Solve Multiobjective Combinatorial Optimization Problems, *MOPGP'96*, Springer-Verlag, 1996.

[7] Goldberg D., *Genetic Algorithms: In Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA, 1989.

[8] Gonsalves, P. and Burge, J., Multi-objective Optimization to Support Rapid Air Operations Mission Planning, *Proceedings of SPIE Vol. #5805*, SPIE Defense & Security Symposium, (Orlando, Florida), 2005.

[9] Gonsalves, P. and Burge, J., Software Toolkit for Optimizing Mission Plans (STOMP), AIAA Intelligent Systems Conference, (Chicago, Illinois), 2004.

[10] Gonsalves P., Burge J., et al., Decision Support System for Theatre Missile Defense. In *Proceedings of SPIE*, (Orlando, Florida), 2003.

[11] Gonsalves, P., Burge, J., et al., *A Decision Support System for TMD Impact Assessment*, National Symposium on Sensor and Data Fusion, (San Diego, CA), 2003.

[12] Jensen, M., Reducing the Run-Time Complexity of Multiobjective EAs: The NSGA-II and Other Algorithms, *IEEE Transactions on Evolutionary Computation* 7(5): (2003) 503-515.

[13] Knowles, J. and Corne D. The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Multiobjective Optimisation, *Congress on Evolutionary Computation*, (Washington, D.C.), 1999.

[14] Knowles, J., Oates, M., et al., Multiobjective Evolutionary Algorithms Applied to Two Problems in Telecommunications, *BT Technology Journal* 18(4), 2000, 51-64.

[15] Mulgund, S., Harper, K., Krishnakumar, K., and Zacharias, G. *Air Combat Tactics Optimization using Stochastic Genetic Algorithms*, IEEE International Conference on Systems, Man, and Cybernetics, October, 1998. (La Jolla, CA), 3136-3141.

[16] Rogers, J., A Parallel Approach to Optimum Actuator Selection With A Genetic Algorithm, *AIAA Guidance, Navigation, and Control Conference*, (Denver, Colorado), 2000.

[17] Ruda, H., Burge, J., Aykroyd, P, Sander J., and Okon, D., *Distributed Course of Action Planning Using Genetic Algorithms, XML, and JMS*. AeroSense, SPIE, (Orlando, FL), 2001.

[18] Schaffer, J., *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*, Vanderbilt University, 1984.

[19] United States Joint Forces Command Joint Warfighting Center, *Commander's Handbook for Joint Time-Sensitive Targeting*, 2002.