Coevolving Collection Plans for UAS Constellations

Daniel W. Stouch, Charles River Analytics[1], dws@cra.com

Ernest Zeidman, Charles River Analytics[1], ewz@cra.com

Marc Richards, Charles River Analytics[1], mdr@cra.com

Kirk D. McGraw, U.S. Army ERDC CERL[2], Kirk.D.McGraw@usace.army.mil

William Callahan, Charles River Analytics[1], bjc@cra.com

ABSTRACT

Our SPARTEN (Spatially Produced Airspace Routes from Tactical Evolved Networks) tool generates coordinated mission plans for constellations of unmanned aerial vehicles by allowing the mission planner to specify the importance of each objective for each mission. Using an evolutionary algorithm-based, multi-objective optimization technique, we consider factors such as area of analysis coverage, restricted operating zones, maximum ground control station range, adverse weather effects, military terrain value, airspace collision avoidance, path linearity, named area of analysis emphasis, and sensor performance. By employing novel visualizations using geographic information systems to represent their effectiveness, we help the user "look under the hood" of the algorithms and understand the viability and effectiveness of the mission plans to identify coverage gaps and other inefficiencies. In this paper, we apply multi-objective evolutionary algorithms to the air mission planning domain, with a focus on the visualization components.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search – *heuristic methods, plan execution, formation, and generation, scheduling.*

J.7 [**Computers in Other Systems**]: Military – *heuristic methods, plan execution, formation, and generation, scheduling.*

General Terms

Algorithms, Performance, Design, Experimentation,

Keywords

Evolutionary algorithms, multi-objective evolutionary algorithms, air operations, mission planning, unmanned systems

1. INTRODUCTION

The dependability, persistence, and versatility of unmanned aerial systems (UAS) have made them indispensable assets for providing intelligence, surveillance, and reconnaissance (ISR) over the battlefield [1]. As larger constellations of heterogeneous, multi-purpose UAS are tasked to perform more diverse missions in unpredictable, dynamic environments, they are transitioning from remote control into the realm of autonomy. One area in which intelligent systems can augment human capabilities is mission planning. As more UAS are employed, the ISR collection routes that they fly must be better coordinated to maximize coverage effectiveness and also take into account complex factors such as sensor performance for specific target types in different terrain with varying military value, as well as weather effects on the platform and sensor. For mission planners to develop confidence in these autonomous mission planning tools, the tools must demonstrate that they can reliably accomplish mission objectives by providing operators with a deep understanding of the capabilities and limitations of their UAS during varying mission scenarios [2].

We are developing a tool called SPARTEN (Spatially Produced Airspace Routes from Tactical Evolved Networks) that generates optimized coordinated mission plans for UAS constellations. In this paper, we present our application of multi-objective evolutionary algorithms to this domain, with a focus on our novel visualization techniques using geographic information systems (GIS). Our visualizations help algorithm developers and end-users "look under the hood" of the algorithms in a domain-specific way to understand the viability and effectiveness of the mission plans, supporting identification of coverage gaps and other inefficiencies.

2. FRAMEWORK

The goal of SPARTEN is to produce a set of optimal flight routes for gathering ISR data over a specific area of analysis. As shown in Figure 1, input data comes from a variety of sources and is represented within the Scenario Model. The Collection Plan Request comes from the mission planner and includes operational mission details such as the location of the area of analysis (AOA), the start and end times of the operation, the location of restricted operating zones (ROZs), and the location of specific named areas of interest (NAIs). The AOA is the polygonal region on the ground where ISR data is being collected, and can be anywhere from a few square kilometers to a few thousand square kilometers depending on the operational scenario. ROZs are regions in 3-D space that should be avoided during specified time windows. NAIs are specific targets or target areas, represented by polygons that are of particular interest during the mission. Each NAI has a specific target type (e.g. dismounted soldier, pickup truck, etc.). Also included in the Collection Plan Request are the location of the other databases and configuration parameters for the Evolutionary Algorithm (EA) such as the mutation and crossover rates. The Air Maneuver Network (AMN), described further in Section 2.1, is a geospatial database containing a planar network whose nodes represent air control points and launch sites for unmanned aerial vehicles (UAVs), and are connected by edges that specify bi-directional transitions between nodes. Sensor performance and weather effect data are updated as new forecasts become available and are attributed on the AMN. Weather effects include details such as where and when inclement weather will have a potentially adverse effect on UAV or sensor performance, and wind speeds at various altitudes. The sensor performance metric provides the probability of detecting different types of targets at specific locations at specific times within the scenario

GECCO'11, July 12–16, 2011, Dublin, Ireland.
Copyright 2011 ACM 978-1-4503-0557-0/11/07...$10.00.

[1]Charles River Analytics
625 Mount Auburn Street
Cambridge, MA 02138
1 (617) 491-3474

[2]U.S. Army ERDC CERL
2902 Newmark Drive
Champaign, IL, USA 61826
1 (217) 373-3328

timeframe. The UAS Data consists of the available aircraft for the mission, their launch sites, and performance characteristics of the various air vehicles, such as flight endurance, operating altitudes, and available onboard sensors. The final inputs are the User Weights, where the mission planner specifies the relative importance of each objective on a scale of 1 to 10. For example, they may decide that Coverage, Sensor Performance, and Military Value are the most important for the current mission and weight them the highest (10). Adverse Weather Avoidance and NAI Emphasis might be secondary concerns and so be weighted with medium importance (5). The remaining objectives may be of little concern and so are weighted as 0 or 1. We describe how these weights are used during optimization in Section 2.2.4. The Cost Factor Maps (CFMs) are concise representations of the mission plan tradeoff spaces that facilitate the rapid computation of fitness values for candidate collection plans. Each CFM contains geospatially and temporally specified data for each edge on the AMN that indicates how that region in space and time is affected by the given concern. Additional optimization factors, such as collision avoidance, coverage, maximum ground control station (GCS) range, latency, and linearity (described in Table 1) are taken into account when the algorithm is run.

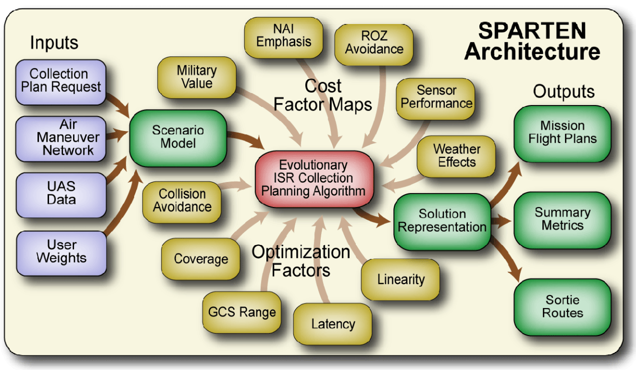


Figure 1. SPARTEN architecture

The CFMs, along with the Scenario Model and the optimization factors, provide all of the data necessary to run the Evolutionary ISR Collection Planning Algorithm, which then generates the Solution Representation. The results within the Solution Representation are presented to the mission planner using the Mission Flight Plans, the Summary Metrics, and the Sortie Routes. Mission Flight Plans are in NATO's Common Route Definition (CRD) format, an XML-based representation of military flight plans. The Summary Metrics are GIS-based representations in ESRI's ArcMap that show how well a given solution performs through the lens of each optimization objective. The Sortie Routes are also GIS-based representations that show the flight routes for reference on a background map of the AOA.

2.1 Air Maneuver Network

The AMN maintains topological and temporal reference frames to represent flyable air corridors for a variety of mission types. In this work, we expand the concept of the AMN presented in [3] to include a topological representation of the AOA that lets SPARTEN treat continuous airspace as a discrete state space. Practically, the AMN is a set of planar networks at various altitudes whose nodes represent air control points and launch sites for UAVs connected by edges that specify bi-directional transitions between nodes. The direction of an edge accounts for effects, such as relative wind speed and sensor performance, which depend on travel direction. A valid route for any asset is a set of ordered edges, where there is a node that connects any two edges. Although an AMN may have planar networks at numerous altitudes, we consider only the single level case in this paper. A representative AMN is shown in Figure 3, where the AOA is a large yellow rectangle, ROZs are polygons cross-hatched in red (Figure 2(a)), and NAIs are polygons hatched in blue (Figure 2(b)). AMN edges are shown as green lines connecting the small green circles that are AMN nodes (Figure 2(c)). The icon in Figure 2(d) indicates where UAS launch and recovery sites are located, and the icon in Figure 2(e) indicates where ground control stations are located.

(a) ROZ **(b) NAI** **(c) AMN** **(d) GCS** **(e) UAS**

Figure 2. AMN symbology

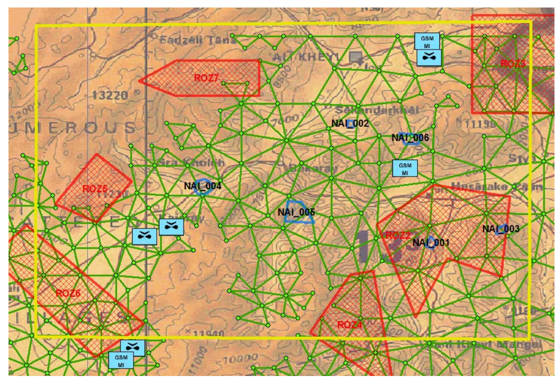


Figure 3. Air maneuver network

Edges represent flight segments and each edge carries a list of attributes relating to flight along that edge for each time interval. These include NAIs, ROZs, weather effects, sensor performance, military value of the underlying terrain, and GCS range. The attributes are calculated individually in advance, based on available information, such as weather forecasts. Each edge also has a length, where the length of the edge corresponds to the traversal distance for the flight segment. Edge attributes are used to evaluate potential solutions during the execution of the Evolutionary ISR Collection Planning Algorithm. No attributes are associated with nodes, since nodes represent instantaneous transitions between edges. However, launch sites, which can also serve as the terminus of an edge, have associated attributes that indicate which type of aircraft can use that resource. A valid route for any asset must start and end at a launch site.

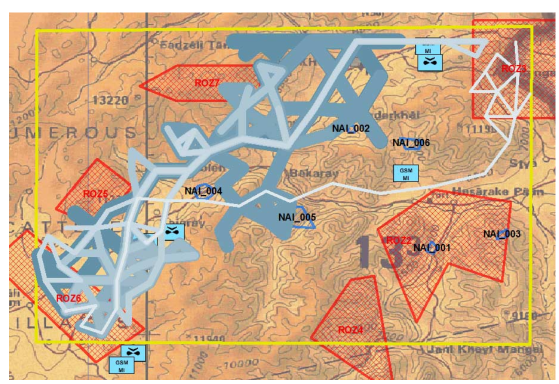


Figure 4. ISR collection route

Figure 4 shows a sample flight route for an ISR mission being flown using that AMN. In this view, the flight route is represented by darker and thicker lines in the beginning of the route that taper to lighter, thinner lines at the end of the route. This mechanism allows overlapping lines to show where the same area has been covered multiple times.

2.2 Optimization Objectives

Each collection plan has multiple objectives against which it is evaluated during the selection phase. The mission planner weights each objective with an integer from 0 to 10 to indicate its relative importance. Each objective's fitness function (y) returns a value of [0, 1], which indicates how well a collection plan (S) satisfies the objective. A fitness value of zero is the lowest possible value, and one is the highest possible value. We describe coverage, sensor performance, and military value in more detail below, but a summary of all the objectives used to evaluate collection plans is listed in Table 1 for reference.

Table 1. SPARTEN optimization objectives

Objective	Description
Adverse Weather Avoidance	Average weather effects value for all the edges in the collection plan, weighted by traversal time
Collision Avoidance	Degree to which the flight routes of all aircraft in the solution avoid each other in time and space
Coverage	Fraction of edges within the area of analysis covered by a UAV within a collection plan
GCS Range	Average ground control station visibility value for all the edges in the collection plan, weighted by traversal time
Latency	One minus the reciprocal of the average number of times each edge is covered by a UAV within a collection plan
Linearity	Measure of the aggregate curvature of all the flight paths within the collection plan
Military Value	Weighted average of the military values associated with each edge that is covered by a UAV within a collection plan
NAI Emphasis	Weighted power mean over the set of average coverage values of each named area of interest
ROZ Compliance	Total time that UAVs are not in restricted operating zones divided by the total time UAVs are in the air
Sensor performance	Average of the probability of detection (PD) values for all the edges in the collection plan

2.2.1 Coverage

The coverage objective measures the number of edges in a collection plan that are completely within the AOA and covered at least once by a UAV. The coverage fitness function y for the collection plan S is calculated as a fraction to map it to the fitness value range of [0, 1], and is given by $y(S)=n/a$, where n is the number of AMN edges covered at least once and a is the total number of edges completely within the AOA.

2.2.2 Sensor Performance

The sensor performance objective measures probability of detection (PD) values for the collection plan. The sensor performance fitness value is calculated as the average of the PD values for all the edges in the collection plan, weighted by the traversal time associated with each edge. Each PD value is dependent on the edge being traversed, the time of traversal (e.g., day, night, twilight), the sensor used (e.g., electro-optical, infra-red), and the target type being sought (e.g., pickup truck, dismounted soldier). The sensor performance fitness function y for the collection plan S is given by the following equation, where K is the number of unique UAVs used in the collection plan, I_k is the number of unique edges in UAV k's flight path, s_k is the sensor being used by the k^{th} UAV, t_{ik} is the time at which the k^{th} UAV is flying over the i^{th} edge, a_k is the target type that UAV k is assigned to observe, $r(s_k,t_i,a)$ is the PD value given s_k, t_i, and a, and Δt_{ik} is the time required for UAV k to traverse the i^{th} edge.

$$y(S) = \frac{\sum_{k=0}^{K}\sum_{i=0}^{I_k} r(s_k, t_{ik}, a_k)\Delta t_{ik}}{\sum_{k=0}^{K}\sum_{i=0}^{I_k}\Delta t_{ik}}$$

2.2.3 Military Value of Terrain

The military value objective measures the military relevance associated with each edge. For example, edges over terrain that is a potential ambush spot or a chokepoint at a crossroads in a mountain pass would have higher intrinsic military value while areas of open desert or ocean would have lower intrinsic military value. The fitness value is calculated as an average, weighted by the traversal time associated with each edge in the collection plan. The fitness function y for the collection plan S is given by the following equation, where K is the number of heterogeneous UAVs used in the collection plan, I_k is the number of edges on UAV k's flight path, m_i is the intrinsic, time-invariant military value in the range [0, 1] of the i^{th} edge in the collection plan, and Δt_{ik} is the time required for UAV k to traverse the i^{th} edge in its flight path.

$$y(S) = \frac{\sum_{k=0}^{K}\sum_{i=0}^{I_k} m_i\,\Delta t_{ik}}{\sum_{k=0}^{K}\sum_{i=0}^{I_k}\Delta t_{ik}}$$

2.2.4 Weighting of Objectives

The mission planner running the SPARTEN engine specifies a weighting value on a scale of 0 (unimportant) to 10 (very important) for each objective listed in Table 1, with the exception of collision avoidance, which is always given a high priority. The fitness values for each objective are multiplied by the normalized weight to incorporate the relative importance of each specific objective in the context of the current mission scenario. The weights are normalized such that they sum to 1.

2.3 Cost Factor Maps

Cost Factor Maps layer on top of the base AMN, including NAIs, ROZs, weather effects, sensor performance, military value of the underlying terrain, and ground control station ranges. We build on the work in [4] by enhancing CFMs to describe the mission plan tradeoff spaces in advance of collection route evolution. This facilitates rapid fitness evaluations during the evolutionary process. Each CFM contains geospatially specified data for each edge on the AMN that indicates how that region in space and time is affected by the given concern. By computing the CFM in advance, we can offload much of the computational complexity involved in fitness evaluation so that it can be done quickly.

Figure 5 shows a sample CFM from SPARTEN for military value of terrain (see Section 2.2.3). The CFM displays in red where the underlying terrain has a high military value, in yellow and orange where it is moderate, and in green where it is low. Figure 6 shows a CFM for sensor performance for a particular type of infra-red sensor on an ISR mission, where there is high probability of detection (in green), moderate PD (in yellow and orange), and low

PD (in red). The region in the lower right of the AOA is particularly poor with regard to PD, as a result of severe weather activity in that area that adversely affects the sensor. Every edge in the AMN is included in the CFM, with the addition of a transit edge between the launch site and the closest node on the AMN. This is in contrast to the solution metric maps (shown below in Figure 11, Figure 12, and Figure 13) that only include edges that are actually traversed by a UAV in the solution. Since coverage can only be computed after a candidate solution has been generated, coverage has no CFM computed in advance.

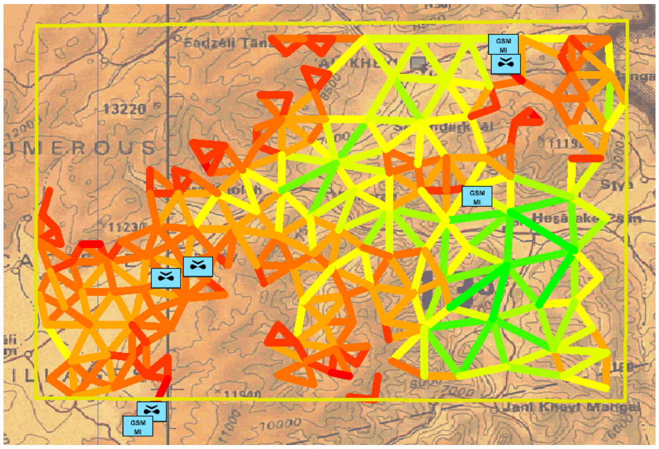


Figure 5. Military value emphasis cost factor map

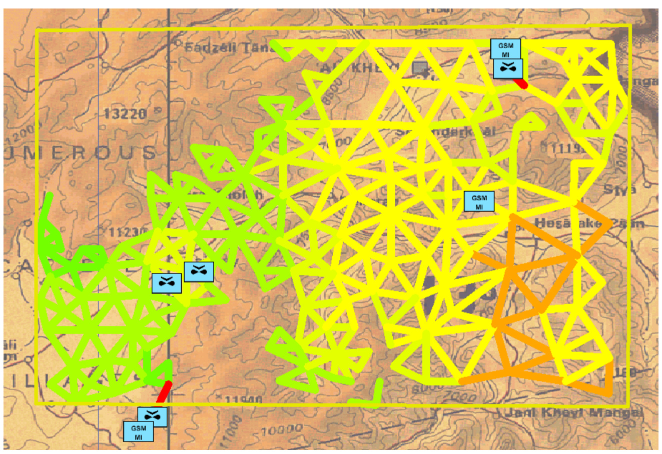


Figure 6. Sensor performance cost factor map

3. ALGORITHM

Each candidate solution consists of a set of flight routes, one for each available UAV. Each flight route is represented by an ordered list of waypoints that corresponds geospatially to nodes on the AMN. Each waypoint has a latitude, longitude, altitude, timestamp, and assigned sensor. This set of flight routes makes up an ISR collection plan, and has a corresponding genetic representation (Section 3.1). The Evolutionary Algorithm maintains and evolves a population of Collection Plans using selection (Section 3.2) and variation (Section 3.3) operators. Crossover operates both on sorties (Section 3.3.1) and individual waypoints (Section 3.3.2). Mutation variation is performed using insertion (3.3.3), and deletion (Section 3.3.4) operators. Our coevolutionary approach is described in Section 3.4.

3.1 Genetic Representation

The collection plan is a coordinated flight plan for a constellation of UAVs. Each of the UAVs in the collection plan flies multiple sorties within the mission. Each sortie has three flight segments, where the linear combination of these flight segments flown by a UAV during all its sorties within a mission is called a flight path.

1. Departure from the launch and recovery site to an AMN node within the area of analysis
2. Mission execution (selection, crossover, and mutation operate only on this segment.)
3. Return from an AMN node to the launch and recovery site

The mission plan individual is a two-tiered structure. Each full chromosome represents a complete flight plan, with each “sortie gene” representing one sortie for one asset. This representation allows sorties to be reordered and swapped among assets via crossover. Each sortie gene is itself a chromosome of “waypoint genes” that specify the individual nodes in the AMN that will be traversed during that sortie. Variation operators on the waypoint genes allow modification to the sequence of waypoints that will be followed during sortie execution. The two-tiered representation extends the approach presented in [5] to support sorties and the AMN concept. The launch times of the sorties in each gene are fixed at evenly spaced points during the length of the mission.

3.2 Selection Operators

The selection operation selects individuals to breed for the next generation. The algorithm assesses each individual against multiple objectives (such as coverage, military value, or GCS range), and then computes one number for each objective, called the individual’s fitness value for that objective. The fitness values of each individual are then compared to determine which individuals are selected. Selection reduces genetic diversity, but promotes a net increase in fitness by weeding out weaker individuals in the population.

SPARTEN implements a two-individual tournament selection operator. The tournament initially applies a sequence of constraint functions, which test validity of both plans. The primary constraint used in SPARTEN is a collision check; a collision is defined as two UAVs occupying the same edge in the AMN at the same time.

Within the tournament, if one of the two mission plan individuals violates the constraints, then the other individual “wins” the tournament and is selected. If both collection plans have violations, then a tiebreak is attempted and the individual with the fewest collisions “wins.” If both individuals are valid, then the selection operator considers their fitness functions.

SPARTEN has one fitness function for each objective. Each objective is weighted by the user to indicate its relative importance. Each fitness function returns a value from 0 to 1, inclusive.

Since the mission plan individual’s overall fitness is determined by its fitness value on multiple objectives, and the distributions of fitness values vary based on the objective, the fitness values returned by these fitness functions must be carefully balanced to be able to compare the fitness of two individuals.

SPARTEN sums the weighted signs of the differences between the individual’s objective fitness values to determine the winner of the tournament. We define T to determine the winner of a tournament between individuals A and B as shown below, where N is the number of fitness functions under consideration. For $1 \le n \le N$, $F_n(X)$ is the fitness value of the n^{th} fitness function applied to individual X, and W_n is the user-specified weight for the n^{th} fitness function. If $(T > 0)$, A is selected. If $(T < 0)$, then B is selected.

$$T = \sum_{n=0}^{N} W_n sign\big(F_n(A) - F_n(B)\big)$$

3.3 Variation Operators

The variation operators modify an individual's genetic material to create new individuals for the subsequent generation. SPARTEN uses two types of crossover operators: sortie operators that exchange sortie genes between assets, and waypoint-level operators that exchange waypoint-based genetic material. The insertion and deletion mutation operators create fresh genetic material at the waypoint level.

3.3.1 Sortie Crossover

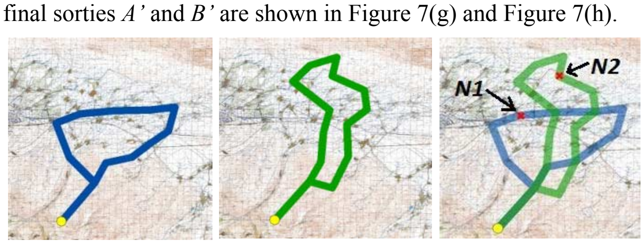
Sortie crossover is performed between two mission plan individuals and involves the exchange of entire sortie genes. The operation is a single-point crossover whereby a sortie from individual *A* is swapped with a sortie from individual *B*. Let *A* be an individual with n sortie genes, a_0 through a_{n-1}. Let *B* be an individual with n genes b_0 through b_{n-1}. The number of sorties in the representation is equal for all individuals, although some sorties may be empty, effectively representing a "no-op."

Crossover between *A* and *B* yields two new individuals, *A'* and *B'*. *A'* has genes a_0 through a_x and b_{x+1} through b_{n-1}. *B'* has genes b_0 through b_x and a_{x+1} through a_{n-1}, where *x* is a random integer from the uniform distribution [*0*, *n-2*]. The probability of any selected individual in the population being involved in a sortie crossover is equal to a user adjustable parameter that defaults to 1.0.

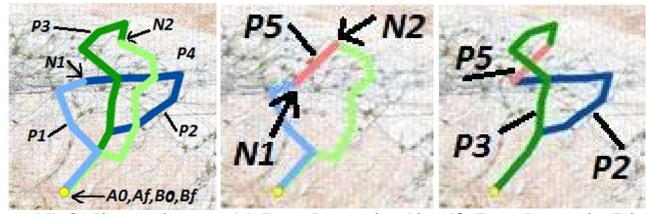
3.3.2 Waypoint Crossover

The waypoint-level crossover operator exchanges genetic information between two sortie genes located in the same position in two mission plan individuals (e.g., the "third" sortie gene or the "fifth" sortie gene in each). The exchange between the genes is done within a single sortie gene, chosen at random. For this example, suppose the "blue" sortie is chosen from individual *A* (Figure 7(a)). The "green" sortie is a symmetric sortie in individual *B* (Figure 7(b)).

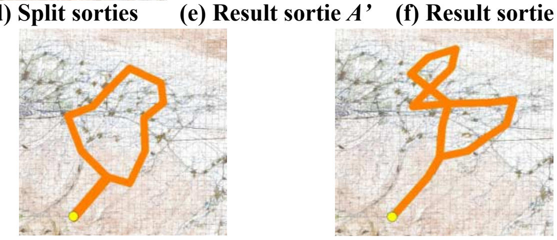
The crossover operator first selects one node from sortie *A* (*N1*) and one node from sortie *B* (*N2*). *N1* is selected randomly from the list of nodes in sortie *A*. *N2* has a distinct, random position. These nodes are shown in red in Figure 7(c). Next, the crossover operator splits the two sorties as shown in Figure 7(d). Let *A0*, *Af*, *B0*, and *Bf* be the start and end nodes of the first and second sortie. In Figure 7(d), these happen to be at the same location but this does not necessarily have to be the case. Sortie *A* is split into *P1*, defined as [*A0, N1*] and shown in light blue, and *P2*, defined as [*N1, Af*] and shown in dark blue. Sortie *B* is similarly split into *P3* and *P4*, respectively defined as [*B0, N2*] and [*N2, Bf*] and shown in dark green and light green. In the third step, a middle section, *P5* (shown in pink), is calculated as the shortest path between *N1* and *N2*. The resultant sorties are *A'* = *P1* + *P5* + *P4* as shown in Figure 7(e) and *B'* = *P3* + *P5* + *P2* as shown in Figure 7(f). The final sorties *A'* and *B'* are shown in Figure 7(g) and Figure 7(h).



(a) Initial sortie *A* **(b) Initial sortie *B*** **(c) Crossover points**



(d) Split sorties **(e) Result sortie *A'*** **(f) Result sortie *B'***

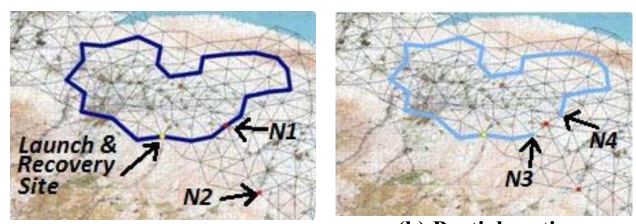


(g) Final sortie *A'* **(h) Final sortie *B'***

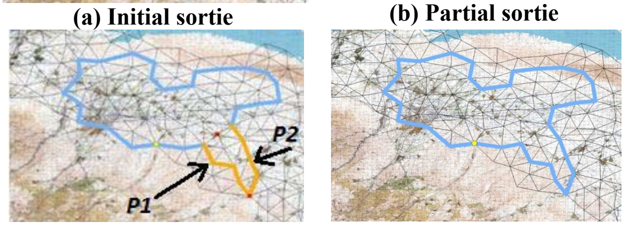
Figure 7. Waypoint crossover

3.3.3 Insertion Mutation

Insertion adds a new flight segment in the middle of an existing sortie. The original sortie is shown in Figure 8(a) with the launch and recovery site shown as a yellow node near the bottom center of the sortie route. The insertion mutation operator selects a node, *N1*, from the list of nodes in the sortie (shown in red along the blue sortie). Another node, *N2*, is selected out of the set of nodes in the AMN (shown in red, down to the right of the sortie). Next, the insertion mutation operator removes node *N1* and its two adjacent edges from the sortie, creating two new terminal nodes, *N3* and *N4*, as shown in Figure 8(b). In the third step, shown in Figure 8(c), the insertion mutation operator creates two new paths, *P1* and *P2*, to connect the sortie to the new node and complete the sortie loop. *P1* is defined as the shortest path between *N2* and *N3*; *P2* is the shortest path between *N2* and *N4*. These new paths are shown in yellow, and the resultant sortie is shown in Figure 8(d).



(a) Initial sortie **(b) Partial sortie**

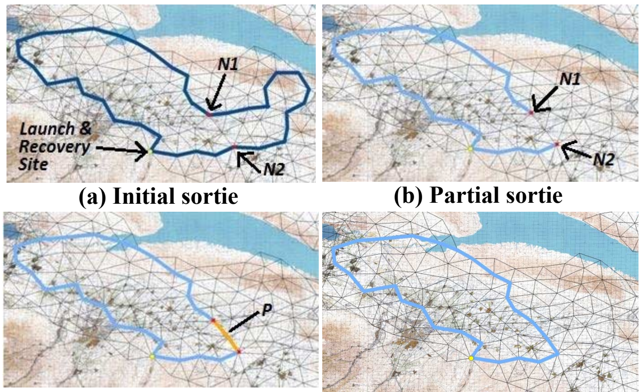


(c) Sortie with new segment **(d) Final sortie**

Figure 8. Insertion mutation

3.3.4 Deletion Mutation

Deletion removes a flight segment in the middle of the existing sortie, replacing it with the shortest-path route between two randomly selected nodes, as demonstrated in the following example. The original sortie within the AOA is shown in Figure 9(a) with the launch and recovery site shown as a yellow node near the bottom center of the sortie route. The deletion mutation operator selects two nodes, *N1* and *N2*, from the list of nodes in the sortie. These nodes are shown in red Figure 9(a). Next, the deletion mutation operator removes the middle section of the sortie defined as ($N1$, $N2$), as shown in Figure 9(b). In the third step, the deletion mutation operator calculates a new path (P) to complete the loop as shown in orange in Figure 9(c). The final sortie is shown in Figure 9(d).



(a) Initial sortie **(b) Partial sortie**

(c) Sortie with new segment **(d) Final sortie**

Figure 9. Deletion mutation

3.4 Coevolutionary Algorithm

We generate results with multiple objectives by employing a coevolutionary algorithm similar to Parmee and Watson [6] where single-objective algorithm instances are used to independently optimize the different fitness functions, resulting in multiple populations of solutions. Each single-objective run starts with a population size equal to a user specifiable parameter (the default is 10), and then evolves until terminated by the user. Selection is determined solely by the one objective function. At the end of each generation the user is presented with the fitness of the current elite individual and two visualizations: the fitness metrics along a Gantt chart and the same metrics geospatially presented on a map of the mission plan. When the user decides that the elite individual is sufficiently fit, the single run is completed.

An aggregate population is formed from the union of a subset of each single objective run to bootstrap the initial population for the next stage. Each subset contains the fittest individuals in its superset. There is always at least one individual in each subset and the number of individuals in any two subsets never differs by more than one. The aggregate population is then evolved over all objectives simultaneously. The selection operator T uses a weighted sum of the signs of fitness differences, Where w_i is the weight associated with the objective function F_i, and the two individuals under comparison are A and B.

$$T(A,B) = \sum w_i \, sgn\,[F_i(A) - F_i(B)]$$

The user no longer has any input on the algorithm, which proceeds as a standard SPARTEN execution for a time specified by a user configurable parameter indicating the number of multi-objective generations to evolve, and is typically between 200 and 5,000, depending on the size of the AOA.

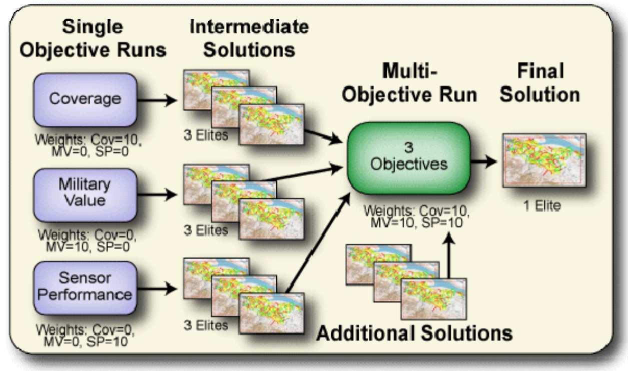


Figure 10. Sample coevolutionary algorithm

For example, a user may want a solution that maximizes coverage, military value, and sensor performance together. The algorithm would proceed as in Figure 10, first optimizing on the three objectives independently. Then the results are combined into a single population to seed a multi-objective run using weighted average of the objective criteria. Newly generated random individuals are also added to the combined population to increase genetic diversity.

In previous work, we had some success applying the NSGA-II algorithm [7] to a similar problem [5]. However, for the present application, the use of the Parmee and Watson-style approach was motivated due to the need to simultaneously optimize as many as 10 objective functions, and NSGA-II's focus on dominance reduces its effectiveness for such many-objective applications [8]. Research suggests combining single-optimization runs may be preferable for such problems (e.g., [9]), although we have not yet explored this research question in our current work. From an end-user perspective, our current approach also presents opportunities for human-on-the-loop preference articulation in which the operator can monitor visualizations similar to those in Figure 11, Figure 12, and Figure 13 (Section 3.5) while the algorithm is executing. When the mission planner determines that the solution for a given objective is sufficient, they can manually accept the result and stop the evolution of that particular objective. We will provide a further analysis of the human-on-the-loop results and implications in future work.

3.5 Fitness Visualizations

Each fitness function results in the computation of a scalar value. While computing a single scalar value for each function facilitates the use of evolutionary approaches, they only provide general information to the user regarding plan quality. To facilitate improved human understanding and evaluation of optimized collection plans, we have developed a variety of geospatial visualizations using ESRI's ArcGIS software. These visualizations show a graphical representation of the AMN over the underlying terrain, using line attributes to represent different aspects of the solution. Figure 11 shows a plan in which the coverage objective has been almost completely optimized. Notice that there are very few AMN edges (from Figure 3) missing in Figure 11. This allows the user to quickly look at the display and see potential gaps in coverage, where they may decide to employ alternate means (e.g., additional special purpose aircraft, satellites, ground personnel) to take up the slack.

This solution was generated after evolving for 300 generations using a population size of 10; one elite; individual crossover probability = 1.0; gene crossover probability = 0.7; and a mutation probability = 0.2. The coverage weight was 10 and the other objectives were weighted at 0.

We have also developed an alternate temporal view of metrics that can be displayed simultaneously with the geospatial information. This visualization is displayed below the geospatial plot in Figure 11. It is a one-dimensional view where time increases to the right. Each colored bar represents the quality of the solution as the mission execution time increases. Each asset is represented by a row, with breaks within the rows for different sorties. The operator may note specific time windows in which an asset is ineffective and manually adjust the plan for those times.

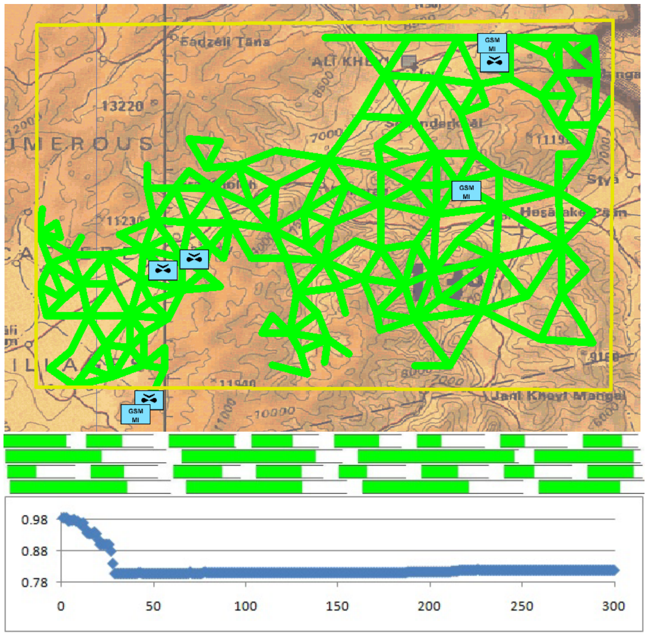


Figure 11. Coverage solution visualizations

The final visualization shows the change in fitness value for the best fit individual from the population after each generation. The initial population starts with relatively high fitness, but these solutions include airspace collisions and are therefore unacceptable. After selective pressure removes the solutions with collisions from the population, the coverage value steadily begins to increase again and converge to the optimized value. A similar phenomenon occurs for military value and sensor performance. Currently, we only visualize these last two when optimizing for a single objective.

Figure 12 shows a similar example using the military value metric. In this visualization, green edges are over regions of terrain with high military value and red edges are over regions with lower military value. Yellow and orange edges indicate intermediate military value. This informs the mission planner of the strengths and weaknesses of the collection plan, and where during the flight routes UAVs are likely to detect targets (green regions) and where their sensors are likely to be less effective. The solution shown in Figure 12 was generated using the same configuration parameters as for coverage, running for 300 generations with the military value weight set to 10 and the other weights set to 0. Figure 13 shows the sensor performance metric for a plan, allowing the user to see what areas are receiving sensor coverage at potentially diminished quality. Again, the same configuration parameters as for coverage were used, running for 300 generations with the sensor performance weight set to 10 and the other objectives all set to zero. Red lines indicate that the quality of coverage along that edge is likely to be dramatically reduced.

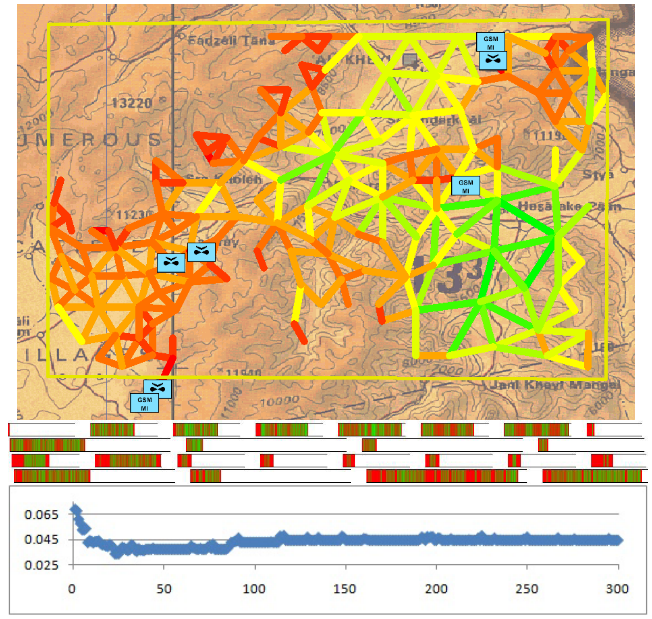


Figure 12. Military value solution visualizations

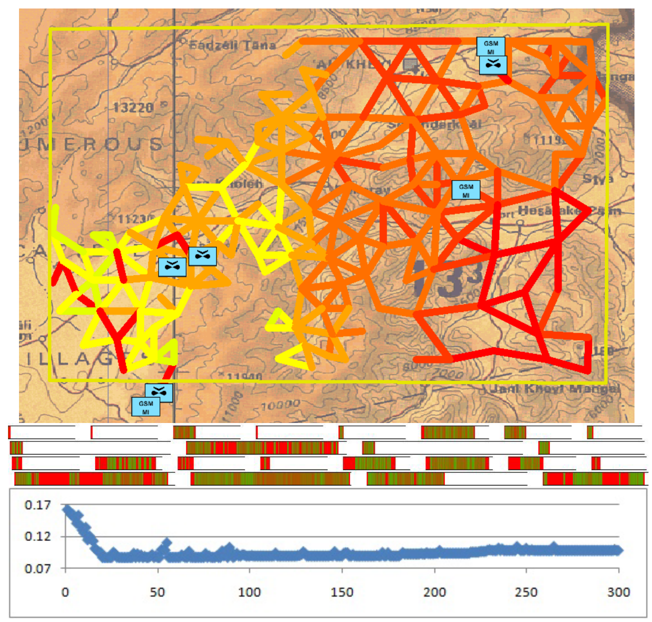


Figure 13. Sensor performance solution visualizations

These visualizations are useful for several reasons. From a research perspective, we can watch them evolve over time to gain better understanding of the algorithm's performance. For example, major changes to the collection plan may result in only small changes to the overall fitness, such that watching raw fitness scores change as the generations progress does not necessarily give an accurate picture of how the algorithm is traversing the search space. In contrast, we can explore the geospatial and temporal visualizations to gain a better understanding of how different parts of the search are explored as the algorithm runs. For example, we can observe that coverage may remain relatively unchanged during destructive crossover (because all the nodes are still being visited as a result of the crossover), but a metric such as sensor performance may be greatly affected because the weather will change as the sorties are executed. We can use these observations to help tune fitness weights and search parameters, such as reducing crossover probability as the algorithm progresses, or reducing the magnitude of mutations.

For the end-user, these visualizations are essential to understanding quality of plans that are returned by the algorithm. The raw fitness values are foreign and somewhat meaningless to a mission planner in charge of managing a UAS constellation. The operational user would much prefer to know precisely where sensor coverage is expected to fail or to know which NAIs are not being effectively covered. The purpose of SPARTEN, after all, is not to provide a fully-automated planning solution that removes the human from the planning process entirely. Rather, SPARTEN is a decision support aid that will provide the human planner with high-quality plans that they can further tweak to meet objectives that cannot be effectively captured in the fitness functions. Our visualizations provide much-needed context and military relevance to the fitness scores, which are a great asset to the human planner.

Figure 14 shows a solution that was generated using the same configuration parameters as for the single-objective runs that was run for 300 generations and takes into account the coverage, military value, and sensor performance weights simultaneously. In this view, the colors represent a normalized average of the weighted fitness values of the objectives being optimized.

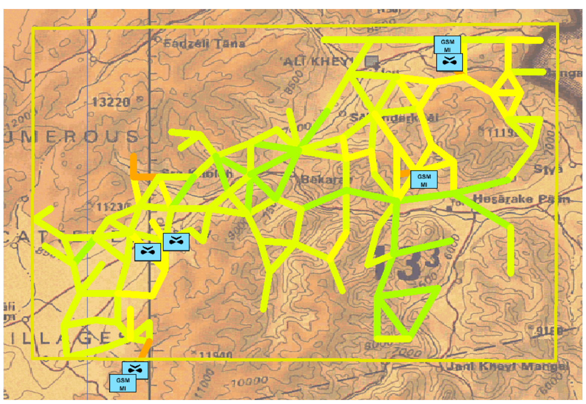


Figure 14. Solution using combined objectives

4. CONCLUSIONS AND FUTURE WORK

We have presented a method to optimize collection plans for constellations of UAVs and provided examples of visualization aids that help the researchers and operational users make sense of the results. Our approach provides a promising decision aid to help mission planners create more effective mission plans for constellations of UAVs by making use of available geospatial and temporal data.

There are a number of potential areas for follow-on research that we are currently pursuing. Primarily, we are interested in using the described visualizations to support improved human interaction with the evolutionary algorithm, as opposed to the current post-hoc analysis. Presenting plans to the user during the evolutionary process does not present a significant technical challenge, although determining effective approaches for performance evaluation of the interactive coevolutionary algorithm remains an open research question. Additional improvements will focus on increasing the operational readiness of SPARTEN. For example, we plan to increase the fidelity of the simulations to support more detailed and accurate flight plans.

We also plan to field test our SPARTEN algorithms using a representative set of small UAVs to gather ISR data over an exemplary AOA. We will also add additional optimization objectives, such as acoustic modeling, to support gathering ISR data while avoiding detection by enemies.

5. ACKNOWLEDGEMENTS

The work presented here has been supported by Government Contract Numbers W9132T-06-C-0040 and W9132T-09-C-0028. The authors would like to thank U.S. Army Engineer Research and Development Center (ERDC) Construction Engineering Research Laboratory (CERL) and Army Geospatial Center (AGC) for their assistance and vision under the Battlespace Terrain Reasoning and Awareness (BTRA) and Geospatially-Enabled Multimodal Situational Awareness (GEMS) programs.

6. REFERENCES

[1] OSD (2009). Unmanned Systems Integrated Roadmap, 2009-2034. Office of the Secretary of Defense.

[2] Taylor, K., Stouch, D., & McGraw, K. (2010). UAS Constellation Routes for ISR Security Operations.

[3] Tenenbaum, S., Stouch, D., McGraw, K., & Fichtl, T. "Multi-Objective Optimization to Support Mission Planning for Constellations of Unmanned Aerial Systems," Proc. SPIE Defense and Security Symposium, 696215-1-696215-12, (2008).

[4] Ablavsky, V., Stouch, D., & Snorrason, M. S. "Search Path Optimization for UAVs Using Stochastic Sampling With Abstract Pattern Descriptors," Proc. AIAA Guidance Navigation and Control, (2003).

[5] Rosenberg, B., Richards, M., Langton, J. T., Tenenbaum, S., & Stouch, D. W. "Applications of Multi-Objective Evolutionary Algorithms to Air Operations Mission Planning," Proc. 2008 GECCO Conference Companion on Genetic and Evolutionary Computation, 1879-1886, (2008).

[6] Parmee, I. C. & Watson, A. H. "Preliminary Airframe Design Using Co-Evolutionary Multiobjective Genetic Algorithms," Proc. Genetic and Evolutionary Computation Conference (GECCO '99), (1999).

[7] Deb, K., Agrawal, S., & Pratap, A. "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II," Proc. Parallel Problem Solving From Nature VI Conference, (2002).

[8] Khare, V., Yao, X., & Deb, K. (2003). Performance Scaling of Multi-Objective Evolutionary Algorithms. In C. Fonseca, P. Fleming, E. Zitzler, L. Thiele, & K. Deb (Eds.), Evolutionary Multi-Criterion Optimization (2632 ed.), (pp. 72). Springer Berlin / Heidelberg.

[9] Hughes, E. J. "Evolutionary Many-Objective Optimisation: Many Once or One Many?," Proc. Evolutionary Computation, 2005. The 2005 IEEE Congress on, 222-227, (2005).