Proceedings of the 2006 Winter Simulation Conference
L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, eds.

GA DIRECTED SELF-ORGANIZED SEARCH AND ATTACK UAV SWARMS

Ian C. Price
Gary B. Lamont

Department of Electrical and Computer Engineering
Graduate School of Engineering and Management
Air Force Institute of Technology
Wright-Patterson AFB, Dayton, OH 45433, U.S.A.

ABSTRACT

Self-organization offers many potential benefits to autonomous multi-UAV systems. This research investigates the use of a self-organization (SO) framework for evolving UAV swarm behavior. This SO framework is used to design a UAV swarm simulation with evolving behavior. The swarm behavior is then evolved using a genetic algorithm (GA) to successfully locate and destroy retaliating stationary targets. This system is tested using both a set of strictly homogeneous UAVs and heterogeneous UAVs with intriguing results.

1 INTRODUCTION

UAVs have significant benefits over traditional aircraft. They can have greater persistence and stamina than human piloted aircraft. Without risk to a human pilot, UAVs can operate in areas exposed to nuclear, chemical, or biological agents. These vehicles can also be sent on extremely dangerous missions without risking human pilots. Finally, removal of the cockpit can improve aircraft performance and reduce design constraints. For these reasons, much of current research examines the ways in which UAVs could best perform. However, automated UAV systems require considerable design to address all possibilities.

One promising answer to UAV automation is self-organization. Multi-UAV systems could successfully function autonomously by harnessing the organizational concepts used by colonial insects, wolf packs, and even economics. Potential for emergent self-organization can also be realized by observing groups of UAVs as a singular macro system rather than a set of interacting individuals.

In Section 2, we introduce a set of features for SO systems and illustrate previous work in multi-agent UAV simulation. Emphasis is placed upon the behavior of those approaches. The design of a SO UAV system and simulation program is discussed in Section 3. The designed system implements a functional SO template created by Price (2006). Experimental testing of the system is described in Section 4. These experiments deal with scenarios where either a homogenous set of UAVs or set made up of both sensing UAVs and UCAVs must engage targets at unknown positions. Section 5 lists the results of those experiments followed by concluding remarks in Section 6.

2 BACKGROUND

This section contains a summary of self-organized definitions and selected UAV swarm systems. Emphasis is on the particular resultant behavior of those systems.

2.1 Self-Organization (SO)

SO applies the principles of organization used by biological entities (ants, bees, herds, bird flocks, etc) to system behavior. According to Camazine et al.. (2003), SO is a systemic approach to unifying multi-agent collections or systems. Systems can be inorganic, however organic biological systems can produce more stunning behaviors.

A set of SO features are distilled and a model is created to facilitate intentional SO design. For more detail see Price (2006). The model states that SO is an attribute of a system with a goal. The system is made up of many lower-level components that interact to produce system wide behavior and performs better than can be achieved by purely individual actions. Lower-level components select their behavior based upon *local* observations without global knowledge of a pattern, strategy or global direction.

2.2 Multi-UAV Modeling

Various investigations address the difficulties associated with UAV deployment and autonomous UAV behavior. In general, the issue is how UAVs select their next behavior. There are two apparent ways in which the this is done: direct control of velocity and rule based directional control.

1-4244-0501-7/06/$20.00 ©2006 IEEE

Direct approaches use a decision making process to determine the turn rate and thrust. This can range from an evolutionary programming mechanism directly encoding each UAV's direction and velocity like that developed by Milam (2003) to a neural network with the outputs tied to velocity and steering as used by Zaera et al. (1996).

Direct approaches can be used to evolve behavior as done by Zaera (1996). Evolving behavior using the direct approach, however, often has limited success. We believe that the reason for these failures can be attributed to the difficulty in evolving too many attributes and behaviors simultaneously. This approach also presumes the designer does not know or assumes that the behaviors they believe are important should be added to the system.

Static systems can also use a direct mapping approach. In this case, the actual behaviors of UAVs are frequently associated with complex mathematical ideas about UAV behavior. The particular appeal of these systems is that they do explicitly what they are designed to do. However, with respect to SO systems, manually created systems are very difficult to construct (Collier and Taylor 2003).

Explicit use of codified behavior rules leads to a different design approach. These rules describe behaviors based upon anticipated needs and compute UAV velocity and heading. For example, Reynolds (1987) described three types of rules for flocking behavior: alignment with neighbors, group cohesion, and close neighbor repulsion.

Behavior rules are not limited to those described by Reynolds. Rather, rules can operate upon anything mathematically describable. In a sense, rules add a complexity layer and predefined behavior. However, they require a suitable combination method like Reynolds' summation.

The different rules can be weighted to afford different resulting behaviors. For example, relaxing repulsion rules can yield better UAV force massing prior to the attack whereas stronger repulsion may improve reconnaissance. Identifying when different weights are beneficial necessitates a decision mechanism, or behavior matrix, such as a neural network like that employed here or state machine as used by Schlecht et al. (2003).

Another improvement can be made by mapping the behavior matrix output to static sets of behavior weights called behavior archetypes as used by Price (2006). This structure has three major effects upon the resulting behavior: behavior matrix complexity can be reduced, rule specific data difficult to represent can be included, and simulation-wide behavior can be more easily understood.

Despite these benefits, the behavior archetype architecture limits rule expression; the rules are not dynamic and operate only as limited static states. If only three behavior archetypes are allowed in a system then only three distinct behavior weighting sets can be defined.

2.3 UAV Sensors and Senses

UAVs rely upon sensor inputs to determine which movements and actions are better than others. The approach taken here is that UAVs utilize a series of sensor values representative of their observed environment, or senses, as input to their movement logic. Senses contain information useful in deciding when and what value each behavioral rule should be applied.

Senses are more refined than explicit distances and vectors as used by Marocco and Nolfi (2005). Since behavior rules already operate upon specific information like positions, velocities, and distances; UAV senses can be more general and abstract without concern over information loss.

A density sense and a target spotted pheromone sense are developed. The density sense is used to determine UAV crowding. This sense has utility by indicating the visible UAV population and their relative distances.

Knowing when targets are spotted is useful for attack coordination according to Kleeman (2004). This sense is used differently than ant-like pheromones (Camazine et al. 2003) – the target spotted pheromone is only calculated at each UAV since tracking environmental propagation is infeasible with real-world airborne UAVs. This sense can create cascading attacks amongst UAVs or specific target avoidance when coupled with specific rules.

In addition to abstract senses, sensor visibility attributes can be explored. When calculating the visible set of UAVs, limitations similar to those in biology have been explored. Parrish et al. (2002) simulated fish with neighbor visibility limited to the closest two or three individuals with low quality results. Kadrovich (2003) uses sensor shadow to allow close UAVs to block visibility of those further away. This approach results in formation stability.

Communication is often implemented in UAV modeling systems. This modeling level is often approached as a flat application of communication device strength like that used by Lua et al. (2003) and Schlecht et al. (2003). This modeling assumes unidirectional broadcast and message reception based upon distance between UAVs. Other works deal with communication in more fidelity. Kadrovich (2003) modeled UAVs as traveling ad-hoc sensor networks with sophisticated communication.

2.4 Exemplar System Models

Math models created by Lotspeich (2003), Kadrovich (2003), and Milam (2004), have been defined. Other important models are those constructed by Schlecht et al. (2003) and Lua et al. (2003).

Kadrovich created his model to study communication and formation stability aspects of UAVs to support operation as a flying ad hoc network. In this model, the UAVs have two major behavior rules: alignment and attraction. These rules encompass those created by Reynolds (1987) with a few implementation differences: Kadrovich relies upon four distinct distances between UAVs to effect swarm cohesion and separation. For example, if two UAVs are within the *too close* distance the attraction rule causes them to separate.

Lotspeich created his model to investigate UAV control. It has more pertinent information of UAV behavior and implements a simple communication system. Lotspeich implemented behavioral rules encompassing cohesion, separation, threat avoidance, and goal seeking. These behaviors are combined by weighted summation of the rules.

Milam's model focuses on the control and behavior of UAVs in a 3D environment. Milam's investigation uses a genetic programming model and does not implement a sophisticated physics model like Lotspeich's structure. However, it relies upon a direct approach to UAV control as the output from the genetic programming component specifically states the actions taken to change the UAV velocity.

The Schlecht model is designed to offer behavior optimized for 2D search by intelligent munitions. In this work, intelligent munitions perform sweeps of a defined area by lining up at a side, and in a parallel formation, search the area while traveling towards the opposite side. If the intelligent munitions locate a target, they determine whether to immediately attack or continue searching.

The Lua model demonstrates sophisticated attacking behaviors. Like the work performed by Schlecht, this model assumes UAV attacks are terminal and that UAVs function as intelligent munitions. This work demonstrates exceptional traits for UAV attack. The UAVs rely upon well defined orbital patterns around the located target and explicit communication to attack with great success.

3 DESIGN

The overall design for the proposed SO system is discussed in this section. First, the mathematical structure for SO is introduced followed by a UAV design implementing the SO features. Finally, operation of the evolutionary algorithm is described.

3.1 SO Model

A system definition for self-organization is composed in order to enable SO development and modeling. A static self-organization definition is reflected in Equation 1.

$$SO \equiv (\sigma_s, \sigma_0, g, \Delta, \tau) \tag{1}$$

In this system, σ_s is the space of all dynamic combinations of possible agents, *A*, and environmental effectors, *e.* These combinations constitute SO system state. σ_0 is the initial start condition for the SO system. Effectively, σ_0 is a state existing within σ_s. The visibility function, *g*, filters which entities within system state affect each agent. Finally, τ, provides a mapping between the agent micro level to the system macro level. The mapping provided by τ allows a separation between individual agent interactions and the resulting emergent system behaviors.

Each SO system state is represented by a dynamic tuple. This tuple accounts for all agents in the system at that time as well as all non-agent effectors. The dynamic tuple representation follows in Equation 2.

$$\sigma (k) \equiv (A_k, e_k) \tag{2}$$

Each tuple represents the set of existing agents in a SO system at a particular state, *k*, as A_k. In addition, the set of effectors, e_k, exists within the dynamic SO state. The function Δ computes the next dynamic state from the previous state as defined in Equation 3.

$$\sigma (k + 1) \equiv \Delta (\sigma (k)) \equiv \Delta (A_k, e_k) \tag{3}$$

The function Δ can operate both synchronously or asynchronously. That is, the update can model changes to all agents or only a subset at each stage.

3.2 UAV Design

A behavior archetype approach is designed and implemented to reduce system complexity and simplify understanding,. The behavior matrix uses a single-layered perceptron to select appropriate behavior archetypes.

Agent or UAV logic is rule-based with system behavior rules. These rules are evasion, obstacle and border avoidance, alignment matching, velocity matching, thresholded cohesion, thresholded separation, weighted target attraction, thresholded target attraction, thresholded target avoidance, unweighted target avoidance, and target orbiting (Price 2006).

This system uses two senses: UAV density and the target spotted pheromone.

The UAV physics model facilitates behaviors identified by the agent logic while maintaining suitable simulation fidelity. UAVs have mass and are subject to deceleration due to drag similar to Lotspeich's model. UAVs are also subject to maximal turn-rates like that in a Dubin's Car, maximum and minimum speeds, and realistic affects of acceleration based upon drag and thrust.

For modeling purposes, UAV capabilities are based upon Predator UAVs as used by Lotspeich. The turn-radius is artificially low since anecdotal experimentation demonstrated less chaotic motion.

When calculating how a UAV should move, the desired next direction vector, obtained by summation of the behavior rules weighted by the active behavior archetype and scaled by the archetype velocity value, is normalized within the UAV's feasible turn rate. This direction vector is then renormalized within UAV physics model and UAV capabilities.

Each UAV has a sensor envelope around its position. Everything outside this envelope cannot be seen whereas UAVs, targets, and obstacles within the envelope are known. Lotspeich suggests that limiting the visible range in this manner allows greater scalability than more global methods. UAVs can passively detect neighboring UAVs' positions, velocities, and pheromonal signals. These values are used by the behavior rules to compute next suggested directions. Additionally, UAVs are subject to sensor shadowing as defined by Kadrovich (2003). This provides formation stability for interacting UAVs.

According to Camazine et al. (2003), communication for SO systems can be split into implicit cues and explicit communication. Implicit cues express information not specifically intended as a communication. UAVs obtain these cues from passive detection of other UAVs as performed by the sensors.

Explicit communications include the implicitly communicated information about a communicating UAV and the location of all targets it can detect. Explicit communications basically provide for one-way sensing; UAVs receiving explicit communication are made aware of communicators position, velocity, and pheromonal signals as well as the location of targets the communicator detects.

UAV simulation states contain information necessary to determine next behavior. These necessary components to agent state include current position, *P*; direction of travel, *D*; behavior archetype, *BA*; target spotted pheromone value, ρ; remaining UAV hit points, *H*; known UAVs, N'; known targets, T'; and known obstacles, O'. These combine to create agent state, s_i.

$$s_i \equiv (P, D, BA, \rho, H, N', T', O')$$

The values of the agent state existing within the SO system macro state are the position, *P*; velocity, *D*; and the remaining UAV hit points, *H*; of each UAV.

Individual UAVs, existing as agents within a SO system dynamic state, update their individual values with the function δ. This update is performed by feeding the sensor values from the environment into the behavior matrix. The resulting values from the behavior matrix indicate which behavior archetype should be used by the UAV. In this way, the behavior archetype *BA* is updated. The position and velocity of the agent are then updated by combining the weighted behavior rules specified by the selected behavior archetype.

The internal neighborhood representation for each agent is generated with the static SO function *g* and the environmental state of the dynamic system. In this case, *g* sorts the dynamic environment and places elements within UAV sensor range in the UAV's local environment representation composed of N', T', and O'.

Actual engagement between the UAVs and targets is modeled using a hit point based system. Experimentation with the system demonstrated that a hit point based model generates more stable results than found in more probabilistic methods. In essence, each UAV and target has a set number of hit points, *H*. During each attack, a UAV or target reduces the hit points of the closest opposing target or UAV within engagement by a damage capacity. Once a UAV or target has zero or less hit points, it is considered destroyed.

3.3 Environment

The environment is the general space in which the system operates. This environment is defined in Equation 4.

$$E \equiv R \times R \times O \times A \times T \qquad (4)$$

In this definition, the environment is a two dimensional space of real numbers, R, with sets of all possible obstacles, O, agents, A, and targets, T.

3.4 GA Design

This system evolves the behavior archetypes and the neural network perceptrons associated archetype selection. Drawing upon inspiration from Marocco and Nolfi (2005), this investigation uses a genetic algorithm (GA) to evolve UAV behavior.

The perceptron, mapping the sensory inputs to behavior archetypes, is fully connected. Connection weights are evolved and each value is represented by 5 bits with a range of [-16 .. 15]. The five bit gene is mapped to its value through a Gray Code in order to facilitate less erratic mutation (Back 2000).

Each behavior archetype rule weight and range is also evolved. These values are also 5 bit numbers. The range for rule expression is mapped through a Gray Code onto the interval [0.0 .. 1.0].

When multiple UAV types exist in the same scenario, the representation increases in size to allot a perceptron and set of behavior archetypes for each UAV type. The genetic structure is demonstrated in Figure 1. There is a connection weight for each sense for each behavior archetype. These are followed by 12 genes which describe the weights and radii for the behavior rules for each behavior archetype.

The GA uses a fixed population size with elitist selection. Despite early convergence, elitism appears highly successful. Premature convergence is not necessarily a concern with this system since solution fitness is not exact. This fitness evaluation inaccuracy may offer results similar to tournament selection method (Back 2000).

This system uses two forms of mutation. The first acts upon the representation as a binary string. This type flips a set number of randomly selected bits within a specified mutation neighborhood.

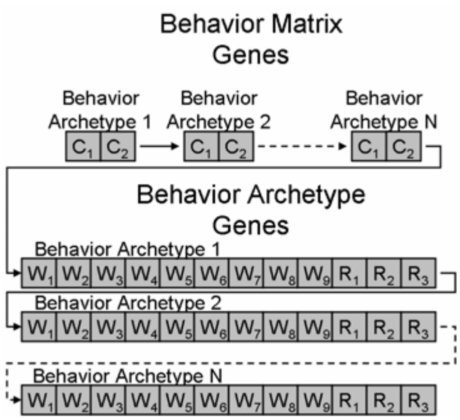

Figure 1: Genetic Structure

The second mutation type, comprising 25% of all mutations, reinvigorates unused behavior archetypes by completely randomizing a selected archetype. This function acts on both the perceptron and the behavior rule values.

A modified two-point crossover recombination is utilized by the GA. With the Gray code representation, points for the crossover are limited to intervals between the 5 bit genes. This prevents the crossover operator from altering the gene values in inappropriate ways according to the Gray code.

In addition to limiting the crossover points, this operator performs 2 two-point crossovers: one within the behavior matrix and the other within the behavior archetypes. This allows crossover within both the behaviors archetypes and the behavior matrix.

Early experimentation demonstrated a large initial learning curve at the beginning of a GA run. Lohn et al. (1999) suggests varied scenario difficulty could alleviate this difficulty. Adaptive scenario difficulty is achieved by changing the specific scenario specifications as the system runs. Anecdotal experimentation with this system suggests a fixed schedule fitness function outperforms a static fitness function.

The fitness of an individual simulation is determined by the amount of damage caused to the targets. This encapsulates the UAV system's need to search the area, coordinate attacks, and successfully damage and destroy targets. The fitness function assigns 10 points for each hit point of target damage.

To prevent overly specific evolution, the scenario initial states are slightly varied when starting each simulation. For this reason, individual simulation scores can vary dramatically. This effect is shown in Figure 2. Composite fitness scores are obtained by averaging the results of a series of simulations and are used for evolutionary purposes. (The vertical line indicates the approximate mean.)

4 EXPERIMENTAL DESIGN

System experiments are meant to develop the *best* SO behaviors for UAVs to search and destroy a number of targets. In this way, the system should address a set of scenarios and evolve well-performing UAV SO behaviors.

There are two scenarios gauging this UAV system's performance and capability. The first scenario considers 10 homogenous UAVs with only implicit communication. The second scenario examines heterogeneous UAVs with explicit communication. These scenarios were created in lieu of existing benchmarks.

System performance is measured by the GA solution fitness improvement across time. This is tracked as an average of the mean scores by generation for each run.

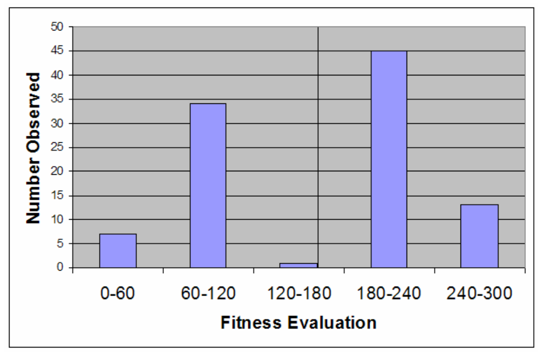

Figure 2: Histogram of Observed Scores for a Particular 49th Generation Heterogeneous Experiment Solution

4.1 Homogenous UAV Experiment

This experiment evolves the best set of behaviors within the systems framework for an engagement between 10 UAVs and 3 targets.

In order to facilitate the experiment, the selected environment is an 80km×80km square with positions described in a Cartesian range of [0 .. 800] by [0 .. 800]. Each individual simulation is run for 3000 simulation seconds (3000 synchronous updates) during which UAVs are updated simultaneously each second.

The 10 UAVs have specific limitations to their abilities. The evolved solutions are limited to three different behavior archetypes. Maximum Sensor ranges are 5km. Explicit communication is not allowed. UAVs can engage targets at 1km distance. UAVs are limited to 77.16 meters per second velocity. Lastly, UAVs have 10 hit points apiece and damage targets at a rate of 1 hit point/second.

Targets, on the other hand, are limited to a different set of characteristics. Only three targets exist in this scenario. The targets have a maximum sensor range of 30km and a maximum engagement range of 2km. They are unable to communicate and are stationary. The targets begin each simulation with 10 hit points apiece and damage UAVs at a rate of 1 hit point per second. Two distinctions between the targets and UAVs are that the targets have twice the UAV engagement range and that targets are stationary.

The UAVs have a centralized starting location with random initial velocities on the north western edge of the environment. Additionally, the starting positions place UAVs within 4.5km to their nearest neighbors. Initial directions are randomly generated to prevent over specialization to the scenario. Coupled with the random initial bearing, the UAVs should, at the beginning of the scenario, already be in some sort of formation since each UAV can sense other UAVs nearby.

The targets have random starting locations placed away from all environment borders. The use of random initial positions for the targets simulates their unknown locations within the environment. Additionally, they are distanced from the borders to force the UAVs to search the interior for targets rather follow environment borders.

The difficulty of the scenario is adapted by varying the engagement range of the targets. This should force the UAV system to learn cooperative attack as the difficulty increases. Table (1) displays the schedule.

The genetic algorithm is run for 50 generations in accordance with the scenario schedule in Table 1. The population for each generation is 100 and the 20 best solutions are carried over between each generation. Both old and new solutions are simulated 50 times to derive fitness scores. The crossover rate is set at 10% and the mutation rate is 90%. The allowed neighborhood for mutation is approximately 5% the solution representation.

Table 1: Adaptive Scenario Qualities

Scenario	Generations	Target Engagement Range
1	0-9	0.4km
2	10-19	0.8km
3	20-29	1.2km
4	30-39	1.6km
5	40-49	2.0km

With these initial positions and vehicle characteristics it seems likely that the UAVs would evolve effective searching behavior within the first 30 generations. The UAVs should use larger spread out formations to locate targets similarly to what is done in the model by Schlecht et al. (2003). In the first 30 generations, since it is possible for a single UAV to destroy a target, it is unlikely that cooperative attack behaviors would be evolved. However, the UAVs should evolve a more cooperative attack strategy bolstering their reconnaissance behaviors in the last 20 generations of each system run. It is expected that this system should evolve behaviors enabling the UAVs to destroy at least 2 targets in each simulation 50% of the time (a resultant mean fitness greater than 150). The system is run 30 times to derive statistically significant results.

4.2 Heterogeneous UAV Experiment

This experiment again uses 10 UAVs against 3 targets. Effectively, the environment, target characteristics, adaptive scenario schedule, and GA values are identical to the first experiment. The differences in this experiment are that that the majority of the UAVs have limited sensing capabilities and are able to explicitly communicate. In this vein, 1 UAV is equipped with a 10km range sensor suite whereas the others are only capable of 1.5km sensing.

Two UAV types with differing populations exist in this scenario: 1 sensor UAV and 9 unmanned combat aerial vehicles (UCAVs). Many capabilities of these aircraft differ from the UAV attributes used in the homogenous UAV experiment. The sensor UAV has a maximum sensing range of 10 km and explicit communication range of 10km. The sensing UAV is unable to attack or damage targets. Correspondingly, the UCAVs have a maximal sensing range of 1.5 km and maximal communication range of 1.5km. The UCAV attack targets with the same effectiveness as the homogenous UAVs. In all other regards, the heterogeneous UAVs are identical to the simulated homogenous ones. It is important to note that since there are two distinct types of UAVs in this scenario, explicit communication should compensate for these differences.

The heterogeneous UAVs also have a centralized starting location. The positions of the UAVs were selected to place the UAVs at the north western edge of the environment and within 1km to each nearest neighbor. Coupled with the random initial bearing, the UAVs should, at the beginning of the scenario, already be in some sort of formation.

Explicit communication should allow improved UAV capability over the first experiment. Since explicit communication allows a UAV to signal both its own traits and the location of other targets, more cooperative behavior should arise. It is expected that the sensing UAV operates in a purely passive reconnaissance role and signals the location of targets to the UCAVs. The UCAVs on the other hand, should stay within the communication range of the sensing UAV and attack communicated targets.

5 RESULTS AND ANALYSIS

For the first experiment, the system displays expected performance with respect to fitness score increases. This can be seen in examination of the plotted mean and best scores for the homogenous experiment (Ho. Mean and Ho. Best) in Figure 3.

Analysis of all individual scores by generation for all runs indicates a reasonable level of predictable performance. This analysis is performed using a Kruskal-Wallis analysis of variance on ordinally ranked scores. Best score examination indicates that the diverse population frequently finds well performing solutions to each scenario within the schedule rather quickly. In contrast, the gradual increases to mean fitness indicate reproduction of better performing solutions throughout the population. Kruskal-Wallis results suggest that the run scores are very similar in performance when the scenario difficulty is increased (see Figure 4). However, the similarity between simulation scores drops when the difficulty increases.

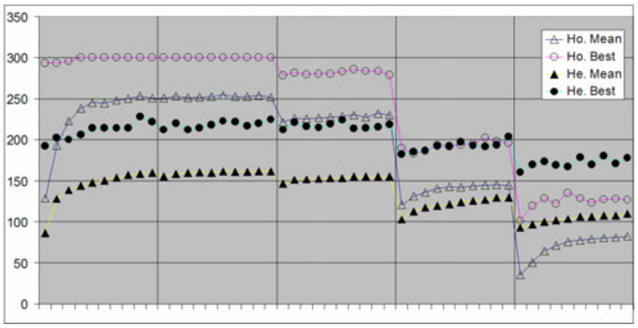

Figure 3: Mean and Best Score by Generation for Both Experiments

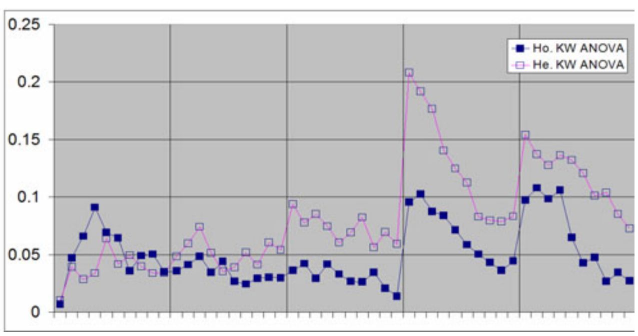

Figure 4: Kruskal-Wallis ANOVA on All Individuals from All Generations

Contrary to expectations, the typical final behavior did not reflect multiple behavior types. Rather, the most frequent solution relied upon a single behavior archetype emphasizing extremely close formations and hyper-aggression. Basically, the UAVs operate in the smallest possible safe formation while attacking targets on sight. Simply put, while the targets have superior engagement range, the UAVs must simultaneous attack a single target for any chance to succeed. If the UAVs were in a larger formation, they would need to shrink their formation prior to attack. In this particular case, it appears that maintaining a smaller formation improves behavior performance even though it diminishes reconnaissance capability.

The final evolved behaviors do not reflect all lower difficulty solutions. For example, an exemplar solution from the 4th difficulty uses two behavior archetypes: one geared towards reconnaissance and the other for attack. While in the reconnaissance mode, UAVs maintain a large formation suited to search and are moderately attracted to UAVs with a high target spotted pheromone. When a UAV detects a target, it switches into the attack behavior and enters into closer formations with cooperating allies.

The performance of the GA with the heterogenous scenario is similar to the homogenous scenario. A major difference, however, is that the maximal fitness score appears bounded by around 230 points whereas the homogenous could achieve the maximal score at many low difficult levels. It is our conjecture that this is cause by disparity in overall sensor coverage between the homogenous and heterogeneous UAV sets. The total area that can be searched at any one time, based upon the combined sensor areas of all heterogeneous UAVs, is about 377.8km^2 whereas the homogenous have about 785.4km^2 out of 6400km^2. It appears reasonable to cite reconnaissance ability as a causal factor for system performance. The statistical results of this experiment can be seen in Figure 3.

When using a heterogeneous combination of UAVs, the resultant swarm must operate both cooperatively when attacking and cooperatively with respect to reconnaissance. For these reasons, the swarm must allow the sensor UAV to perform reconnaissance and the UCAVs to cooperatively destroy targets. Typical solution behaviors are similar to those expected but do not perform as well as anticipated. The UCAVs prefer being in a tight formation centered upon the sensor UAV. The small formations make target avoidance by the sensor UAV difficult in most cases. However, it seems that many successful solutions include the ability for the sensor UAV to remain outside the target engagement range while guiding the UCAVs for the attack. An example of this phenomena is shown in Figure 5 which is a snapshot from one of many animations.

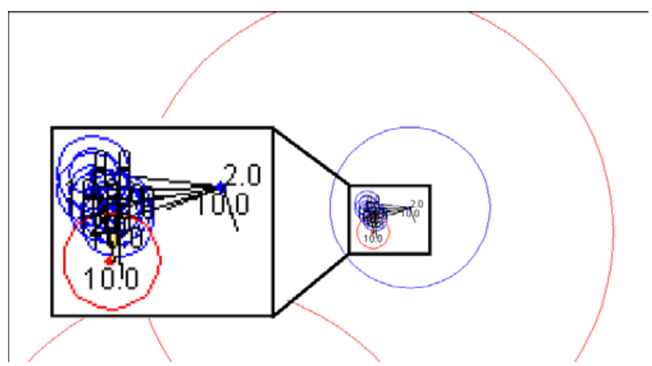

Figure 5: Demonstration of Target Avoidance by Sensor UAV

Overall, the heterogeneous swarm reconnaissance is sub-par. When searching for targets, UAVs blaze a winding path through the environment. The overall best score for this scenario appears bounded by the reconnaissance capabilities of the sensor UAV. Since there is no possibility of organized search patterns with this system's lack of reconnaissance rules, the most effective solutions rely upon brute sensor coverage. This behavior appears universal to all well scoring solutions for this experiment.

6 CONCLUSION

This work has significance in three ways: it provides a SO model, engagement between retaliating targets and UAVs is modeled, and it demonstrates an architecture for future modeling and SO UAV swarm behavior evolution.

The self-organized model we created for this system demonstrates results similar to other works modeling attacks upon targets like that completed by Lua et al. (2003) and by Schlecht et al. (2003). Implicitly, those works address the features of self-organization without explicitly using a self-organized model.

Systems modeling directed attacks upon targets from aircraft appear quite common. However, investigations modeling retaliating stationary targets are not found in the literature. The exact engagement model applied here is very abstract and assumes a quantifiable amount of hit points can be assessed to both UAVs and targets for tactical engagement. Though abstract, the hit point approach incorporates UAV attrition and better models the need for behavior flexibility and robustness.

The behavior archetype model developed provides for necessary behavior evolution when considering particular scenarios. This system could be used to represent and develop UAV behavior for many potential scenarios. In addition, the rule-based approach supports future inclusion of additional rules governing such behaviors as reconnaissance. Though the evolved behaviors were not quite as effective as expected, they did demonstrate surprising qualities. For example, it was seen that a smaller formations, though diminished in reconnaissance capabilities, are often favored when the targets have greater engagement range.

With respect to the design and use of a SO model for successful UAV operation, it appears to have been successful through simulation. The SO model allowed for the evolution of a multi-agent system which, in most cases, conferred a cooperative and useful set of UAV behaviors.

ACKNOWLEDGEMENTS

The sponsor of this research is the Air Force Research Laboratory Information Directorate and the Sensors Directorate. We also thank AFIT Prof. Bert Peterson for his role in motivating a consistent and complete SO model.

REFERENCES

Back, T., D. B. Fogel and T. Michalewicz, 2000, *Evolutionary Computation 1 and 2*, Institute of Physics

Camazine, S., J.-L. Deneoubourg, N. R. Franks, J. Sneyd, G. Theraulaz, and E. Bonabeau. 2003. *Self-Organization in Biological Systems*. USA: Princeton University Press.

Collier, T. C. and C. Taylor. 2003. *Self-Organization in Sensor Networks*. Tech. Rep 1618935. UCLA Department of Organismic Biology, Ecology, and Evolution.

Kadrovich, T. 2003. *A Communications Modeling System for Swarm-Based Sensors*. Ph.D. dissertation. Air Force Institute. of Technology.

Lohn, J. D., G. L. Haith, S. P. Colombano, and D. Stassinopoulos. 1999. A Comparison of Dynamic Fitness Schedules for Evolutionary Design of Amplifiers. In *Proceedings of the First NASA / DoD Conference on Evolvable Hardware.*

Kleeman, M. P. 2004. Self-organization. Tech. Rep. Air Force Institute of Technology.

Lotspeich, J. T. 2003. Distributed Control of a Swarm of Autonomous Unmanned Aerial Vehicles. Master's thesis. Air Force Institute of Technology.

Lua, C. A., K. Altenburg, and K. E. Nygard. 2003. Synchronized Multipoint Attack by Autonomous Reactive Vehicles with Local Communication. In *Proceedings of the 2003 IEEE Swarm Intelligence Symposium*.

Marocco, D. and Nolfi, S. 2005. Emergence of Communication in Embodied Agents: Co-Adapting Communicative and Non-Communicative Behaviours. In A. Cangelosi et al.., eds., *Modelling Language, Cognition and Action: Proceedings of the 9th Neural Computation and Psychology Workshop*. Singapore: World Scientific.

Milam, K. 2004. Evolution of Control Programs for a Swarm of Autonomous Unmanned Aerial Vehicles. Master's thesis. Air Force Inst. of Tech.

Parrish, J., S. Viscido, and D. Grunbaum. 2002. Self-Organized Fish Schools: An Examination of Emergent Properties. *Biological Bulletin*, no. 202, pp. 296–305.

Price, I. 2006. Evolving Self-Organized Behavior for Homogeneous and Heterogeneous UAV or UCAV Swarms. Master's Thesis. Air Force Institute of Technology.

Reynolds, C. W. 1987. Flocks, Herds, and Schools: A Distributed Behavioral Model. *Computer Graphics 4*, M. C. Stone, Ed., vol. 4. SIGGRAPH. pp. 25–34.

Schlecht, J., K. Altenburg, B. M. Ahmed, and K. E. Nygard. 2003. Decentralized Search by Unmanned Air Vehicles using Local Communication. In *Proceedings of the International Conference on Artificial Intelligence*, vol. 2.

Zaera, N., D. Cliff, and J Bruten. 1996. (Not)Evolving Collective Behaviors in Synthetic Fish. *Fourth International Conference on Simulation of Adaptive Behavior.*

AUTHOR BIOGRAPHIES

IAN C. PRICE is the chief of high performance computation at AEDC, Bldg 100, suite A325704 CS/CL, Arnold AFB, TN 37389. He graduated from Air Force Institute of Technology with a Master's degree in 2006. His e-mail address is <ian.price@arnold.af.mil>.

GARY B. LAMONT is a professor in the Department of Electrical and Computer Engineering, Graduate School of Engineering at the Air Force Institute of Technology specializing in evolutionary algorithms and high performance computation. He is an IEEE Life Senior Member. His e-mail address is <gary.lamont@afit.edu>.