Fuzzy Ant Colony Algorithm for Terrain Following Optimization

Brett Taylor
iAccess Technologies, Inc.
Warner Robins, Ga, USA
brett.taylor@iaccesstech.com

Anthony Choi
Electrical & Computer Engineering
Mercer University School of Engineering
Macon, GA, USA
choi_ta@mercer.edu

***Abstract*— Ant colony optimization is a class of swarm intelligence algorithms used to solve combinatorial optimization problems. Fuzzy ant colony optimization adds fuzzy logic to combine heuristic data used in the algorithm. In this paper, a fuzzy ant colony algorithm is applied to a class of path planning known as terrain following optimization. Terrain following optimization attempts to find paths through which terrain can be used to avoid line of sight based detection. The algorithm is tested against both controlled test cases and real world terrain. The results show the algorithm effectively balances terrain masking and path length to create flyable terrain masking paths.**

Keywords- terrain masking; terrain following optimization; ant colony; fuzzy logic

I. INTRODUCTION

Terrain following is a mode of flight during which an aircraft maintains a fixed altitude or set clearance plane above the terrain. In a technique known as terrain masking, military aircraft use a low set clearance plane to exploit rugged terrain and remain hidden from detection [1]. The U.S. military now extensively uses high-flying reconnaissance and surveillance systems in multiple missions. This development program's goal is to produce demonstrations of next-generation unmanned flight systems that are designed to perform combat missions. Surviving the mission is the ultimate goal for these unmanned aerial vehicles. Knowing their terrain and flying as fast as possible will greatly improve their effectiveness in first-strike combat operations [2, 3].

Historically, terrain following systems relied exclusively on radar to provide real time terrain data. The problem arises when the pilot is flying low over an unknown terrain; after the next hill, there is no information about the terrain from the radar. Technological advancements have allowed secure flight trajectories to be calculated if adatabase of the terrain is available. The only drawback of this database is that it contains no information about vertical obstructions such as trees or buildings because the reference height is set at 50 meters. The optimal system uses the terrain database to calculate a global trajectory in real time and the radar to closely follow the terrain while detecting and avoiding vertical obstructions. The system's output can be presented to the pilot for manual flight or coupled directly to an autopilot for autonomous flight [4, 5].

In the early 1990's, research methods of optimal trajectory path planning for terrain following and masking was researched. From this, a reduced order formulation based on a constant velocity approach for airplanes was born. Now, a more realistic approach is to use a constant energy formulation. This better accounts for the abilities of the vehicle [6]. The research on these two methods: constant velocity vs. constant energy has spiked an interest in determining the best way to achieve maximum utilization of the terrain following systems.

Terrain masking is one of the functions used in a number of Air Force electronic mission planning systems. This process is used to determine where an electronic emitting device, like a radar, has a line of sight for a particular range or field of view [7]. Terrain masking is divided into two classes: intervisibility and hidability. Intervisibility is the use of terrain masking to hide from known threats; while hidability refers to the use of terrain masking to minimize the chance of detection from unknown threats [8].

Proper path planning is essential to the optimal utilization of terrain masking [1]. Path planning for terrain masking requires minimizing a multivariable cost function that combines terrain masking, path length, and flyability. Algorithms designed to automate terrain masking path planning are called terrain following optimization algorithms [8].

Previous research has focused on algorithms such as Dijkstra's, dynamic programming, rapidly exploring random tree, mixed integer linear programming and genetic algorithms for terrain following optimization [1, 9]. Many of the above algorithms are capable of minimizing the terrain masking cost of a path but do not inherently generate flyable paths. Instead, either the user must manually modify the path or the algorithm is rerun until a flyable path is generated [1].

In this paper, a fuzzy ant colony algorithm is applied to the terrain following optimization problem. One advantage of the ant colony algorithm is that flight dynamics can be directly incorporated into the algorithm so that solutions created by the algorithm are flyable. Another advantage is that fuzzy ant colony algorithms explicitly include heuristic information in the formation and evaluation of a solution. The ant colony algorithm is tested against real world terrain to examine its feasibility as a terrain following optimization algorithm.

II. Ant Colony Algorithms

Path planning to find efficient vehicle routes has been studied for over 40 years. Many different algorithms have been analyzed in order to achieve best results for the vehicle routing problem. Introducing additional constraints such as time windows in which services must be performed, requirement of inserting both pickup and deliveries of goods, to the presence of more than one depot greatly increases the complexity of the problem. To achieve better results, literature and research is increasingly focused on heuristic and metaheuristic approaches. A metaheuristic approach is a computer method that optimizes a problem by trying to improve a solution while also analyzing quality [24]. Another example deals with protein folding. This is by far one of the most challenging problems in biology and physics. This consists of finding the functional shape in a 2-D or 3-D space [25].

Ant colony optimization (ACO) is a class of swarm intelligence algorithms designed to solve combinatorial optimization problems first introduced in 1991 by Dorigo, Maniezzo, and Colorni [10-12]. ACO operates iteratively by combining a priori information about the structure of a solution with a posteriori information about the structure of previously obtained solutions. Through this process, the ACO utilizes two sets of heuristic information: constructive heuristic and local search. The constructive heuristic starts from a null solution and adds elements to build a complete solution while a local search modifies some of the elements of a complete solution to achieve a better solution [11].

Ant colony algorithms draw inspiration from the foraging behavior of ants that use pheromone deposits to find the shortest path between a food source and their nest [13]. The artificial ant system also incorporates features that are not found in their natural counterparts. For example, local heuristic functions give artificial sight that helps guide individual ants in the formation of their solutions. Memory is also used by preserving global best solutions for use in future iterations [19].

Using ants' social behaviors to analyze paths taken is still a relatively new approach and it exploits a similar mechanism for solving optimization problems [14]. However, ant colony algorithms have been successfully applied to many different optimization problems including the travelling salesman problem [13, 15], quadratic assignment problem [16], scheduling [17], reservoir operation [18, 19], network routing [20] and path planning [9, 21, 22].

Ant colony algorithms operate by letting a population of ants search the graph representing the solution space of the problem. Individual ants are more likely to take paths on which previous ants have deposited pheromone giving more recent ants the ability to learn from the fitness of solutions generated by previous ants.

After all ants have formed a solution, the solutions are graded by a heuristic function. Pheromone relative to an ant's grade is applied to the graph based on the ant's path. The entire pheromone map is then reduced by a percentage in a process known as pheromone evaporation which helps prevent convergence on local optima [10]. The solution forming, grading, and pheromone depositing process is repeated for a fixed number of iterations.

III. Fuzzy Ant colony algorithm

Fuzzy logic is a framework through which heuristic knowledge can be mathematically quantified. Among other advantages, fuzzy logic has the ability to model nonlinear functions without exact knowledge of the proper interaction between function parameters. A fuzzy ant colony system uses fuzzy logic to integrate heuristic data into the ant colony framework. Fuzzy logic can be used in both the local heuristic function and in path grading for pheromone depositing.

A survey of literature indicates that fuzzy ant colony systems perform well relative to those using crisp data sets or other heuristic algorithms. Salehindejad and Talebi [22] used a fuzzy ant colony system to perform multiparemeter route selection for vehicle navigation. In addition to route distance, a fuzzy function was created that included the amount of traffic and historical risk of accidents on a given path. The authors found that the fuzzy logic based ant colony system outperformed traditional ant colony systems and ant colony systems guided by the A* algorithm.

Ginidi, Kamel, and Dorrah [16] studied a fuzzy ant colony system incorporating fuzzy logic in both construction of ant solution and the pheromone updating process. For construction of each ant solution, fuzzy logic was used to combine heuristic parameters to select incremental ant steps. The pheromone map was divided into two quantities: a crisp pheromone value and a fuzzy probability density function relating to the amount of uncertainty of the pheromone value. In applying their fuzzy ant colony system to the travelling salesman and quadratic assignment problems, the authors found that the inclusion of fuzzy logic in the ACO algorithm improved results over non-fuzzy ACO implementations.

IV. Implementation

Ant colony algorithms are most often applied to routing or scheduling problems in which all nodes in the problem space are to be traversed or path finding among a limited number of nodes [22] or possible moves [9]. The solution sought in this paper will be capable of path finding on a large graph where only a small fraction of the nodes are traversed in any given solution. Unique to this problem is that a small deviation from established pheromone trails can lead an ant to never re-establish connection with the pheromone trails. The result is that later iterations are unable to refine already discovered paths. Several adjustments are made to the ant colony algorithm to accommodate this issue and are noted as the implementation is described.

The ant colony algorithm used in this paper mixes ideas from several ant colony algorithm implementations. Two ideas from the MAX-MIN ant colony algorithm, first proposed by Stutzle [26], are used: pheromone limiting, in which pheromone levels are limited between an upper and lower bound, and including the global best solution in each iteration. The MAX-MIN Ant System (MMAS) is one of the best performing algorithms for the traveling salesman problem and quadratic assignment problem. The results of the study show that even the worst results were within 0.3% deviation from the optimum traveling salesman problem. Borrowed from the Ant Colony System [12] is the pseudorandom proportional rule which biases ants towards previous solutions. The results of this study show that ants choose the best path when less iterations are analyzed. With 25 iterations, less than 0.4% of the ants didn't choose the shortest path.

A. Ant Dynamics

The current state of an ant is composed of two values: the latitude and longitude pair describing the ant's position and the ant's heading. Maneuver-based motion planning [9] is used to build each ant's solution. Every state transition three candidates are generated representing the new state if the ant makes a unit step along its current heading or a heading change by $\pm 45^0$. If a candidate state requires a vertical change greater than ±300ft, it is removed from the candidate list to impose real world vertical constraint of air vehicle flight dynamics. If all three states are removed, the ant is aborted and not used in calculating the pheromone map.

A pseudo-random proportional rule (1) including exploitation and biased exploration is used to select a new state. Exploitation transitions the ant to the best candidate state and is guided by a uniformly distributed random number q. The parameter q_o determines how likely exploitation is to occur. When not using exploitation, the ant uses biased exploration where each candidate state has a weighted probability of being selected.

$$P(M_i) = \begin{cases} \frac{\tau_i^{\alpha} \cdot {}_i^{\beta}}{\sum_i \tau_i^{\alpha} \cdot {}_i^{\beta}} \;\; if\ q > q_o \\ 1\ if\ \ q < q_o\ and\ i = \arg\max\left\{\tau_{ij}^{\alpha} \cdot {}_{ij}^{\beta}\right\} \\ 0\ if\ \ q < q_o\ and\ i \neq \arg\max\left\{\tau_{ij}^{\alpha} \cdot {}_{ij}^{\beta}\right\} \end{cases} \qquad (1)$$

In (1) τ is the pheromone level and ή is the tour construction heuristics grade. The parameters α and β are used to allow weight assignment to the pheromone and tour construction factors. In this paper, α = 1 and β = 2 are used. The tour construction heuristic is a fuzzy function that combines terrain masking and state heading relative to heading to target to evaluate the candidate states. The surface of the one-step look ahead fuzzy function is shown in Fig. 1.

To force refinement of already discovered paths q_o is increased in later iterations. For the simulations in this paper, q_o is initialized to 20 and increased for each iteration to a maximum of 95. For iterations 75 and later, the result is a 95% chance that the ant utilizes exploitation and not biased exploration. The typical path length being studied is in excess of 100 state transitions so at q_o = 95 each ant typically still deviates from the best path multiple times.

B. Pheromone Updating

The pheromone trail is updated every iteration once all ants have completed a solution. The global best solution from all previous iterations is included in the list of completed solutions. The completed solutions are then compared against each other using a fuzzy function that evaluates the average terrain masking and path length of the solutions. The surface of the fuzzy function used for path grading is seen in Fig. 2.

In many applications, ant colony algorithms apply pheromone to the edges connecting two nodes in the graph because the edges represent the cost function attempting to

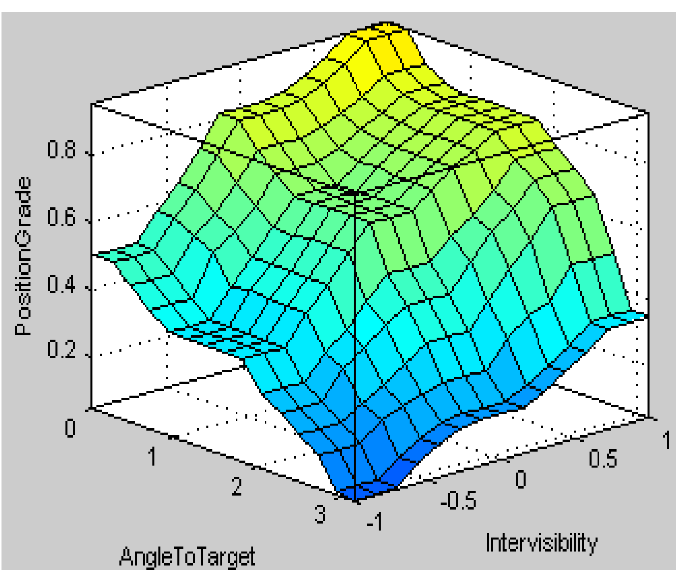

Figure 1. Fuzzy tour construction function

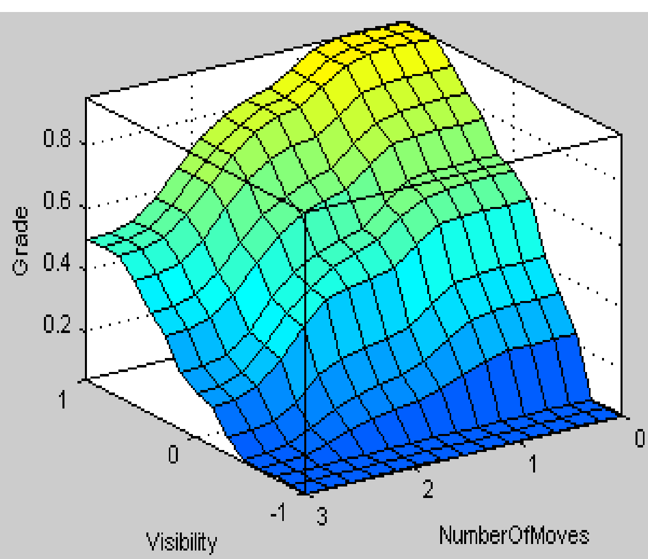

Figure 2. Fuzzy path grading function

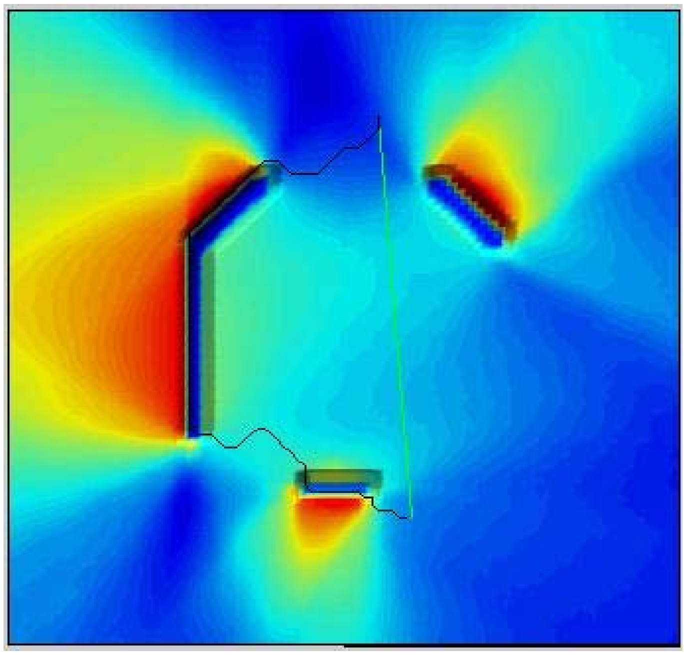

Figure 3. Example test map result

be minimized. For terrain masking, the primary cost lies with the masking value associated with each node and not the edge used to reach the node. To account for nodes possessing the primary cost, pheromone was applied to the node instead of the edge used to reach it.

For each iteration the top 5 solutions had pheromone applied to their path using (2).

$$\tau_{ij} = \begin{cases} 0.9 \times (\tau_{ij} + \frac{0.3}{k}) \; if \; \tau_{ij} \in S_k \\ 0.9 \times \; \tau_{ij} \qquad\quad if \; \tau_{ij} \notin S_k \end{cases} \qquad (2)$$

In (2) τ_{ij} is the pheromone at node (i,j) and S_k is the nodes composing the k[th] solution. Typically areas with high terrain masking scores will be grouped together. To exploit this knowledge pheromone equal to half the pheromone applied to the primary node was spread to adjacent nodes. Spreading pheromone increases the likelihood that path refinement occurs.

V. RESULTS

The ant colony algorithm was tested using a series of created maps and real terrain. For each scenario a hidability map was created using the assumption that the aircraft maintained a constant 200ft set clearance plane. Every node on the hidability map was given a value equal to the number of other nodes for which line of sight was available at 200ft above that nodes terrain. In the included figures the masking maps are overlaid on the terrain as heat maps. Areas with red offer good hidability while areas that are blue offer bad hidability.

Initial testing of the ant colony implementation utilized custom maps created to provide controlled scenarios where results could be visually verified. The custom maps included a flat earth and a map created with multiple structures to create a varying but controlled terrain masking field. These maps were used to test proper operation of the algorithm and to allow parameter tuning in a controlled environment.

The flat earth scenario is a direct analogue to an aircraft flying over water. Flat terrain offers no terrain masking advantage so the ideal solution is a straight line path. The ant colony algorithm was tested with a flat earth scenario that had a straight line path length of 84.85 unit steps. Over fifty separate runs the algorithm produced an average solution length of 87.03 unit steps with a standard deviation (STD) of 0.31.

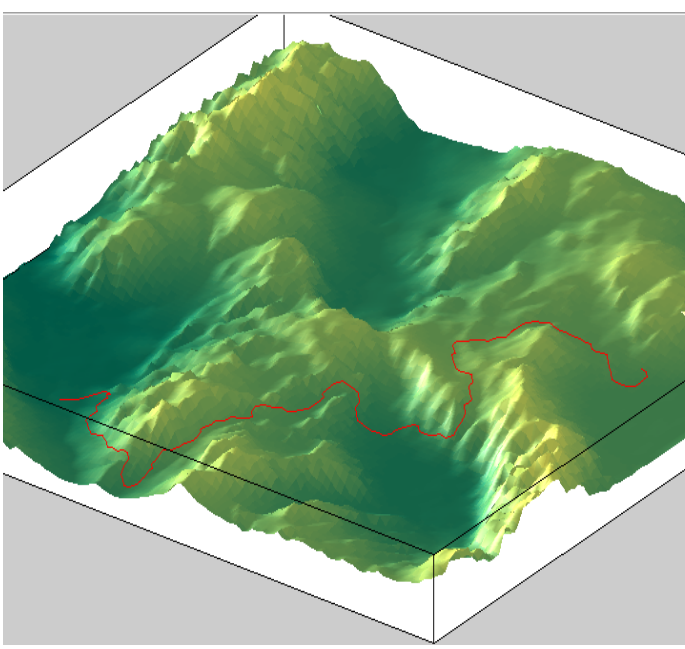

Figure 5. Category 1 topographical map

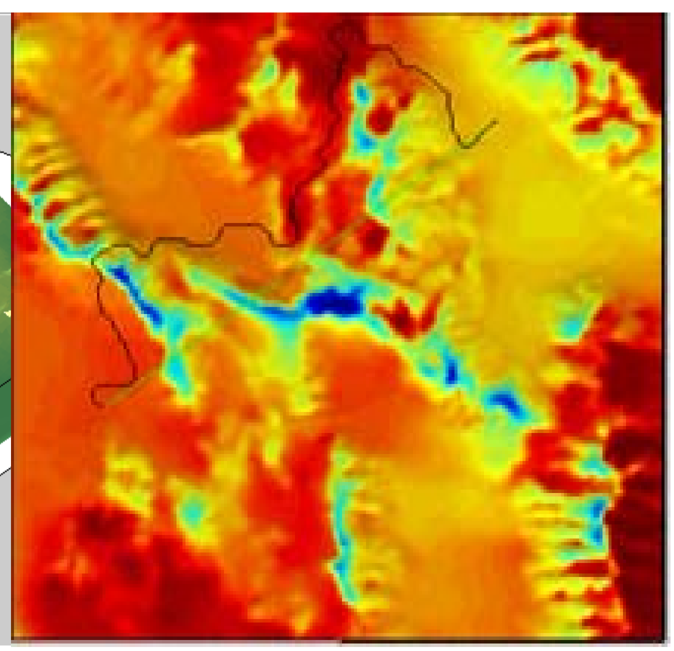

Figure 6. Category 2 example solution

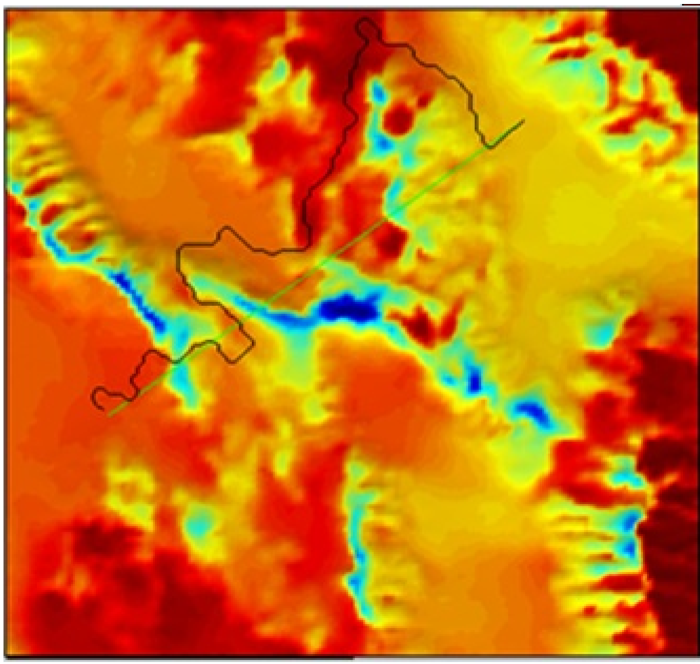

Figure 7. Category 3 Solution

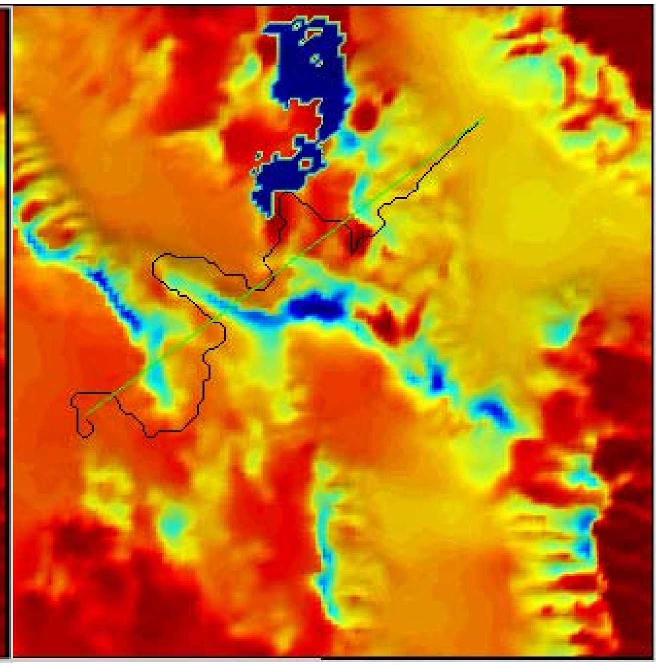

Figure 8. Example solution with threat avoidance

The multiple structures map is shown in Fig. 3. The start (top of map) and end point (bottom of map) for the path are at either end of the straight green line. From the starting point an area of good hidability is located behind the structure to the right. However better overall performance requires the algorithm to find and follow the structures to the left side of the map. The black line shows a solution created by the ant colony algorithm which finds and closely follows the left hand structure in what can visually be verified as a suitable path.

After tuning and testing in controlled scenarios, the ant colony algorithm was tested on one degree latitude by one degree longitude map whose top left corner is located at W118, N46. The map was created using Digital Terrain and Elevation Data Level 0 (DTED 0) data as would be used in a real application. DTED is a grid of elevation values for the given terrain for which three levels are available: level 0, 1, and 2. Level 0 is the lowest resolution with each 1x1 degree cell featuring 120x120 data points spaced approximately every 900 meters.

Thirty runs of the algorithm were tested using a starting position of W117.25, N35.83 (top right) and an ending position of W117.85, N35.46 (middle left). Solutions were evaluated based on two criteria: the solution length and the hideability score. The hideability score was determined by the average number of nodes from which the aircraft was hidden as it traversed the path. When known threats were included solutions were also visually verified to not pass through the threat's line of sight.

The straight line solution for this test case had a path grade of 4011.1 and a path length of 66.6 km. The fuzzy ant colony algorithm on average produced a 21% better solution with an average path length 2.18 times longer than the straight line length. The balance of path length to hidability could easily be parameterized by creating a variable fuzzy path grading function. A summary of the results is found in Table 1.

TABLE I. RESULTS SUMMARY

	Grade	**Path Length (km)**
Average	4856.5	145.02
STD	19.241	6.14
Maximum	4910.9	161.1
Minimum	4828.3	129.6
Spread	81.7	31.5

Visually, the solutions fell into three categories. Fifteen of the thirty results fell in the first category. These solutions gave the best terrain masking grade. An example category one solution is shown in Fig. 4. This particular solution had a score of 4868.4 and a path length of 151.2 km. A topographical map of this solution is shown in Fig. 5.

Eight of the thirty results fell into the second category. While the upper half of the path is similar to the first category the lower half of the path took an alternate route. However, both the score and path length of this category of solution were very close to the first category. An example solution from category 2, which had a score of 4861.7 and a path length of 143.1 km, is shown in Fig. 6.

Seven of the thirty results were obviously suboptimal. An example is shown in Fig. 7 which had a grade of 4838.8 and a path length of 147.6 km. While most of the path is acceptable, near the end of the route the algorithm routed the aircraft over a short stretch of terrain with poor hidability which was avoided in category one solutions.

The ability of the algorithm to avoid threats was tested by adding known threats into a portion of the map preferred by all solutions described in the previous results. A binary intervisiblity map was created for the threat by determining any points on the map at which the threats would have line of sight to 200ft above the terrain. The hidability and intervisibility maps were merged to create a single terrain masking map by assigning a value equal to the worse hidability score squared for visible positions in the intervisibility map and then adding the two maps together.

Over multiple tests the algorithm succesfully avoided the threats while still exploiting hidability in the terrain. An example solution featuring threat avoidance can be seen in Fig. 8. The added threats force the algorithm to take a different route than was seen in the previous solutions. After changing the route to avoid the threats the bottom half of the solution is very similar to the category one solutions.

VI. Conclusion

Terrain following optimization is a challenging problem for path finding algorithms. Heuristic information is required to balance terrain masking and path length. The algorithm must also balance aircraft physics to create flyable solutions. In this paper, a fuzzy ant colony algorithm was created and applied to create an optimized solution. Fuzzy ant colony algorithm is a natural framework for terrain following optimization. Fuzzy logic models the heuristic data and tour construction can be limited to inherently create flyable paths.

References

[1] R. L. Broderson and D. G. Lewis, "An evaluation of terrain following/terrain avoidance/threat avoidance its potential payoffs and technology needs," AFWAL-TR-86-1024. Air Force Wright Aeronautical Laboratory, 1986.

[2] S. Twigg, A. Calise, and E. Johnson, "On-line optimization including moving threats and targets," AIAA Guidence, Navigation, and Control Conference and Exhibit, pp. 1-12, August 2004.

[3] T. Lapp and L. Singh, "Model predictive control based trajectory optimization for nap-of-the-earth (NOE) flight including obstacle avoidance," Proceeding of the 2004 American Control Conference, vol. 1, pp. 891-896, July 2004.

[4] C. Flood, "Real-time trajectory optimization for terrain following based on non-linear model predictive control," Linköping University, vol. 1, pp. 1-61, November 2001.

[5] A. Lucia, P. DiMaggio, and P. Depa, "A geometric terrain methodology for global optimization," Frontiers in Global Optimization, vol. 1, pp. 1-18, 2003.

[6] S. Twigg, A. Calise, and E. Johnson, "On-line trajectory optimization for autonomous air vehicles," American Institute of Aeronautics and Astronautics, vol. 1, pp. 1-9.

[7] M. Benincasa, "Parallel implementation of the terrain masking algorithm," Rome Laboratory, vol. 1, pp. 1-67, March 1994.

[8] G. B. Lamont, "UAV swarm mission planning development using evolutionary algorithms and parallel simulation – part II," RTO-EN-SCI-195, NATO Science and Technology Organization.

[9] T. Krenzke, "Ant Colony Optimization for Agile Motion Planning," Master's Thesis, Massachussets Institute of Technology, 2006.

[10] M. Dorigo, V. Maniezzo, and A. Colorni, "Positive feedback as a search strategy," Dipartimento di Elettronica, Politecnico di Milano, Italy, Tech. Rep. 91-016, 1991.

[11] M. Doringo, M. Birattari, and T. Stützle, "Ant colony optimization: artificial ants as a computational intelligence technique," University Library of Bruxelles, vol. 6, pp.28-39, November 2006.

[12] R. Kumar, M. Tiwari, and R. Shankar,"Scheduling of flexible manufacturing systems: an ant colony optimization approach," Institution of Mechanical Engineers, vol. 217, pp. 1443-1453, June 2003.

[13] H. Maier, A. Simpson, A. Zechin, W. Foong, K. Phang, H. Seah, and C. Tan, "Ant colony optimization for design of water distribution systems," Journal of Water Resources Planning and Management, vol 129, pp. 200-209, June 2003.

[14] Y. Bin, Y. Zhen, and Y. Baozhen, "An improved ant colony optimization for vehicle routing problem," European Journal of Operational Research, vol. 196, pp. 171-176, 2009.

[15] M. Dorigo and L.M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," IEEE Transaction on Evolutionary Computation, Vol. 1, No. 1, April 1997, pp. 53-66.

[16] N. Sabharwal and H. Sharma, "A recursive ant colony algorithm for the TSP," International Conference on Advancements in Information Technology, IPCSIT Vol. 20, 2011.

[17] A. Ginidi, A. Kamel, H. Dorrah, "Development of new fuzzy logic-based ant colony optimization algorithm for cominatorial problems," Proceedings of the 14th International Middle East Power Systems Conference, Cairo University, Egypt, 2010.

[18] P. Mathiyalagan, S. Suriya, S. Sivanandam, "Modified ant colony algorithm for grid scheduling," Internation Journal of Computer Science and Engineering, Vol. 2, No. 2, 2010, pp. 132-139.

[19] D. Kumar and M. Reedy, "Ant colony optimization for multi-purpose reservoir operation," Water Resources Management, Vol. 20, 2006, pp. 879-898.

[20] M. Dorigo, G. Caro, L. Gambardella, "Ant algorithms for discrete optimization," Artificial Life, Vol. 5, 1999, pp. 137-172.

[21] F. Lucas, C. Guettier, P. Siarry, A. Milcent, A. Fortelle, "Constrained navigation with mandatory waypoints in uncertain environment," Information Sciences and Computer Engineering, Vol 1, No. 2, 2010, pp. 75-85.

[22] H. Salehinejad and S. Talebi, "Dynamic fuzzy logic-ant colony system-based route selection system," Applied Computational Intelligence and Soft Computing, Vol. 2010, Article ID 428270.

[23] Y. Bin, Y. Zhen, and Y. Baozhen, "An improved ant colony optimization for vehicle routing problem," European Journal of Operational Research, vol. 196, pp. 171-176, 2009.

[24] P. Pellegrini, D. Favaretto, and E. Moretti, "Multiple ant colony optimization for a rich vehicle routing problem: a case study," University Ca' Foscari of Venice, vol. 1, pp. 1-31, 2006.

[25] C. Blum, "Ant colony optimization: Introduction and recent trends," Physics of Life Reviews, vol. 2, pp. 353-373, October 2005.

[26] T. Stutzle and H. Hoos, "MAX-MIN Ant System," Journal of Future Generation Computer Systems, Vol. 16, 2000, pp. 889-914.