

Randomized Path Planning with Deceptive Strategies

Philip Root, Jan De Mot and Eric Feron
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA 02139
{proot, jdemot, feron}@mit.edu

Abstract—In this paper we study a non-cooperative zero-sum game where one player performs reconnaissance while the second player constantly observes the first. This game has implications for teams of UAVs operating within aural and visual detection range of threat forces. In particular, the threat can potentially react dynamically to UAV observations and endanger future movements. We propose a specific behavior essential to an optimal policy for a team of agents, and create a randomized algorithm inspired by these heuristics. Implementation of this path planner onto a team of autonomous helicopters demonstrated the utility of the algorithm in real time applications.

I. INTRODUCTION

In recent years, the development and fielding of new unmanned aerial vehicle (UAV) platforms has outpaced the development of novel doctrine concerning their implementation. In particular, small UAVs (SUAV) operate largely independent of each other, and there is no established strategy for the use of multiple SUAVs to perform existing missions. Most interestingly for the purposes of this paper, SUAVs now operate within aural and visual detection ranges of the enemy, and hence interact with the enemy more directly[1, 13, 15]. This tactical shortcoming allows a dispersed observer to gain a credible advantage over a reconnoitering opponent.

The future of combat will be marked by increasingly non-contiguous warfare where no clear delineation between opposing forces exists[8]. While history abounds with examples of such similar combat operations such as the U.S. experience in Vietnam, the Balkans, and Somalia, the Russo-Chechen conflicts exemplify this trend toward non-contiguous operations and in particular for operations in urban environments[14]. We will adopt our model of the opposing forces largely from observations of these historical conflicts. To this end, we will refer to the belligerents as defenders and invaders as we introduce the problem, and more simply, as observers and agents as we formulate the problem and present a solution.

In all of the above historical examples, defenders used a markedly similar command architecture emphasizing tactical dispersion. The defenders array themselves in a dispersed arrangement that does not offer the invaders a centralized target to attack. They rarely appear outwardly as combatants and hence are indistinguishable from the local population. Spread uniformly throughout the city, these observers seek to capture as much intelligence as possible pertaining to the invader's intentions and actions.

The defenders operate in largely independent cells while reporting sensitive information to a centralized headquarters via redundant means of communication[10]. This headquarters collects these intelligence reports to identify any trends in the observations; any trends or correlations within the reports may identify the invader's future movements. If they deem the intelligence sufficiently credible, the defenders will establish a threat or ambush at the location of maximum likelihood of contact. In a dense urban environment, this distributed arrangement with centralized planning offers the defenders a much more complete analysis of the current situation than often available to the invader. Additionally, the thorough observer dispersion makes it improbable that the invader can maneuver without being observed[12].

The invader possesses roughly the opposite advantages and attributes. Composed of a large number of soldiers, vehicles, and materiel, the invader enjoys numerical and weapons superiority. This structure, however, makes it impossible for the invader to operate without being observed. Isolated in more secure areas of the urban environment, the defender must routinely move through areas observed by the defender for offensive or logistical reasons. These convoy movements are highly susceptible to threats, and the invader takes every effort to ensure their security. Before deploying along an intended convoy route from an initial position to the objective, the invader identifies critical areas that can influence this movement. While some of these areas may be likely ambush sites along the intended convoy route, others may be known or predicted defender locations. Reconnaissance of these critical areas in addition to observing the intended convoy route allows the invader to ensure the security of the convoy as much as feasible[6, 7].

A relevant study in non-cooperative games is presented in [9] where a defender controls the information available to an attacker. If the defender is allowed full manipulation of the information, it is optimal for both players to randomize their strategies from within the set of zero information alternatives, and deception offers no advantage. Alternatively, if the defender is afforded only partial control of the information, the attacker can gain a credible advantage if its sensors are reliable; otherwise, the defender's use of deception has effectively rendered the available information useless. The deception strategy presented is applicable to a large range of two-player games to include pursuit-evasion games, negotiation, and card games. Research in

randomized choreography in [4] illustrates the utility of a randomized optimization model that satisfies a set of constraints while maximizing surprise. This paper expounds upon this foundation by suggesting a mixed strategy as related to deception efforts when the invader has partial control of information. Specifically, randomizing the observations available to the defender is an advantageous deception policy for the invader.

In light of the relevant work above and the scenario outlined previously, we propose the following simplified game to introduce the goals of both belligerents. Player one, invader, begins by selecting a reference path from a set of all feasible paths from source to objective and deploys a team of agents to reconnoiter the chosen path. Player two, defender, observes these agents, identifies any trends within these reported observations, and decides whether to establish an ambush, and if so where the most probable location of intercept. If player two creates an ambush along the reference path, it wins; otherwise player one wins. It is clear that mixed strategies for both players are optimal, *i.e.* both players choose randomly with a uniform probability distribution from its options.

We formally present this problem in Sec. II and propose a heuristic based solution in Sec. III. We conclude with brief statistical analysis of the proposed algorithm as well as implementation on a team of autonomous helicopters.

II. PROBLEM FORMULATION

In this section, we introduce notation pertaining to the environment and threat model, and formally present the problem at hand as a two-player game.

A. Environment and Player Models

We assign a vertex to each street intersection and to positions along the streets so that the distance between neighboring vertices is of similar size. Let v_i represent the i th vertex and let $V = \{v_i\}$, for $i = 1, \dots, n_v$. Let v_i and v_j (for $i, j \in \{1, 2, \dots, n_v\}$) be a pair of neighboring vertices, *i.e.* the geographical location associated with these vertices are connected by a street (e.g. a pair of neighboring intersections). Then, for all such vertex pairs, we have the undirected edge e_{ij} connecting v_i and v_j such that $e_{ij} = e_{ji}$. Let E_1 denote the set of such edges. We model the urban environment as an undirected complete graph, $G = (V, E)$, with E equal to the union of E_1 and E_2 , where E_2 is the set of edges that makes G a complete graph. In particular, E_2 contains all the edges e_{ij} connecting vertices $v_i \in V$ with $v_j \in V$ that are not in E_1 and, hence, do not correspond to a street in the urban environment. Furthermore, to each edge $e_{ij} \in E$ is associated a cost t_{ij} . For $e_{ij} \in E_1$, the cost is the time required to travel between the geographic locations associated with the vertices v_i and v_j along the connecting street. For $e_{ij} \in E_2$, the cost is time required to travel the length of the straight line connecting v_i and v_j . This distinction allows for path planning with an aerial vehicle not constrained to follow streets.

We define a set of edges $E_m \subset E_1$ associated with the streets that support military traffic. For example, for a heavy vehicle to traverse a particular street, the latter must satisfy certain minimum specifications such as width, radius of curvature and load-carrying capability. In this case, military vehicles are constrained to travel those edges $e \in E_m$.

Furthermore, given is a set of vertices $V_A \in V$, which are associated with critical urban areas (see Sec. I), where threat detection ahead of time is essential for the success of the mission. To each $v_k \in V_A$ is associated the time t_{v_k} it takes a UAV to gather sufficient information at that location.

We define a *path* p of length n_p as a sequence of edges $p = \{e_1, e_2, \dots, e_{n_p}\}$, $e_1 = (v_1, v_2)$, $e_2 = (v_2, v_3), \dots, e_{n_p} = (v_{n_p}, v_{n_p+1})$. The cost associated with the path p is represented by

$$t_p = \sum_{e_{ij} \in p} t_{ij} + \sum_{v_k \in p} t_{v_k},$$

where $v_k \in V_A$ is the start or end vertex of some edge in p . The path $p_r = \{e_i^r\}$, for $i = 1, \dots, n_r$, the *reference trajectory*, is a special path with initial vertex v_s and final vertex v_t , and such that $e_i^r \in E_m$, for $i = 1, \dots, n_r$. The main task of the agents is to determine whether p_r is threat-free by observation. In particular, every edge of p_r is traversed and observed by at least one agent.

We assume that the opponent has a set of n_o observers that can be located anywhere in the urban environment. While these observers can have any spatial distribution in general, in this paper we assume the worst case where they can observe every edge $e \in E_m$ and every node $v \in V_A$. We further assume that these observers communicate spatial and temporal information on agent detections to a centralized decision making unit. This headquarters seeks to determine the reference trajectory, p_r , given the observations of the agents' reconnaissance (see Sec. I). We assume that the central unit must make two fundamental decisions: whether to establish a threat or ambush, and, if so, where to locate this ambush. Let r be a feasible path from v_s to v_t , and let R be the set of all feasible paths, $R = \{r\}$. Let \mathcal{X} be a discrete random variable representing the reference trajectory that the agents select. Then P_r is the probability that path r is the selected reference trajectory for all $r \in R$ such that $P_r = P(\mathcal{X} = r)$. In our model of this central unit, we assume that the headquarters must establish with a certain probability threshold, c_t , that an edge belongs to the reference trajectory before deciding to establish an ambush. A given edge, e , may belong to multiple feasible paths, r , and hence, the probability that $e \in p_r$ is $P_e = \sum_{r: e \in r} P_r$. Note that while $P_e < 1$ for all $e \in R$, the sum of P_e for all $e \in R$ is greater than unity because there are multiple edges in every feasible path. If the maximum probability over all edges, $P_e^* = \max_{e \in R} P_e$, is less than this threshold, (*i.e.* $P_e^* < c_t$), then the headquarters does not establish an ambush. If $P_e^* \geq c_t$, then the headquarters places the ambush at the edge with the highest probability

in the reference trajectory, $e^* = \arg \max_{e \in R} P_e$. Let q be the probability that an observer detects an agent and communicates this observation to the central unit.

B. Formulation as a Two-Player Partial-Information Zero-Sum Game

We formulate the problem at hand as a two-agent partial information zero sum game as follows. The first player consists of the set of N agents, the second player is the opponent, with the capability to gather information and organize one threat in the area of operations.

Given a particular reference trajectory p_r , the possible strategies for the N agents (the first player) are any set of trajectories $P = \{p^k\}$, where $p^k = \{e_i^k\}$ and $e_i^k \in E$, with associated cost t_{p^k} for $k = 1, \dots, N$ satisfying the following constraints:

- Given T_{max} , a maximum allowed agent trajectory cost, we require that $t_{p^k} \leq T_{max}$. The former is determined from the particular vehicle platform, and is mainly derived from fuel and battery constraints.
- For each edge $e \in p_r$, there is an agent k such that $e \in p^k$. In words, each edge of the reference trajectory is traversed by at least one agent.
- For each vertex $v \in V_A$, there is an agent k such that v is the start or end vertex of an edge $e \in p^k$. In words, all vertices in V_A are visited by at least one agent.

The game is then played as follows. The first player chooses a set of agent trajectories, and the agents traverse their trajectories, gathering the required information on the reference trajectory and on the vertices in V_A . The second player collects information on the geographic location and time an agent observation has taken place and determines the probability, P_e , that each edge e is a member of the reference trajectory. Player two then determines the edge e^* with the maximum likelihood of belonging to the reference trajectory with associated probability P_{e^*} . If $P_{e^*} < c_t$, player two does not establish an ambush; if $P_{e^*} \geq c_t$, player two places the ambush at e^* .

The first player wins and the second player loses in case $e^* \notin p_r$ or if $P_{e^*} < c_t$. If $P_{e^*} \geq c_t$ and $e^* \in p_r$, the second player wins, and the first player loses. We look for the optimal, possibly mixed strategies, for both players.

III. PROBLEM SOLUTION

In this section, we argue a set of heuristics to determine the agents' optimal strategy to the game outlined in Sec. II-B. We then suggest a randomized algorithm that creates feasible trajectories and evaluates them against the threat model.

A. Heuristic Strategies

The optimal strategy for player one is not immediately apparent. We propose that the optimal policy is to traverse as many edges as possible within the set E_m with a random sequence and orientation. We present this conclusion as the result of the following induction.

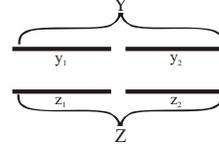


Fig. 1. Simplified environment to develop optimal strategy.



Fig. 3. $T_{max} = 3$ and player two has imperfect information, $q < 1$

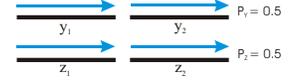


Fig. 2. Player two has perfect information, $q = 1$

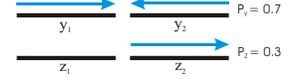


Fig. 4. $T_{max} = 3$ and player one uses random sequence and orientation

Fig. 1 depicts a highly simplified environment to recreate the problem formulation above. There are four edges, $E_m = \{y_1, y_2, z_1, z_2\}$, that form two feasible paths, $Y = \{y_1, y_2\}$ and $Z = \{z_1, z_2\}$. Let R be the set of these paths, $R = \{Y, Z\}$, from the source on the left to the objective on the right. Player one must choose one path $r \in R$ as the reference trajectory, p_r . We assume the worst case where player two arrays observers at every edge $e \in E_m$. We simplify the agent dynamics such that T_{max} refers to the number of edges that player one can visit $p = \{e_i\}$ for $i = 1 \dots n_e$ where $n_e = T_{max}$. We disregard any travel time required to pass from one edge to another, and we assume that player one has only one agent without loss of generality.

Let \mathcal{X} be a discrete random variable that represents the feasible reference trajectories. Player two then determines the probabilities $P_Y = P(x = Y)$ and $P_Z = P(x = Z)$ based the information available from the set of agent observations. We need not determine the probabilities for each edge, P_e , as discussed in Sec. II because each edge only belongs to one feasible path r , and hence, $P_{y_1} = P_{y_2} = P_Y$. Again, q refers to the probability that an observer detects an agent on an edge.

In Fig. 2, $T_{max} = 4$ and player two has perfect information concerning the agents' trajectory such that $q = 1$. This results in $P_Y = P_Z = 0.5$. The consequences are two-fold: if $c_t \geq 0.5$, then player two will not establish an ambush; if $c_t < 0.5$, then player two will establish an ambush, but must choose a route $r \in R$ uniformly.

Fig. 3 illustrates the case where player two has imperfect information concerning the agents' trajectories, $q < 1$, and cannot determine the resources available to player one. We argue that P_Z is non-zero because there exists probability $(1 - q)$ that player two did not observe an agent on edge z_1 .

Finally, Fig. 4 illustrates the scenario when the route is non-sequential; in this example $p = \{y_1, z_2, y_2\}$. [6] states that this non-sequential and mixed orientation of trajectories along path Y may reduce the probability along that path P_Y .

From these simple exercises, we can develop a reasonable strategy for player one. We observe that it is ideal to traverse all edges $e \in E_m$ such that player two must choose a route

$r \in R$ with a uniform probability distribution. Additionally, it is beneficial to randomize the sequence and orientation among all links traversed. We create a set of *deception routes*, E_d , that contain all edges in E_m and not in p_r such that $E_d \subset E_m \setminus p_r$.

B. Path Generation

Each path set $P = \{p^k\}$, for $k = 1, \dots, N$ with N available agents is created as follows. We create n_r disjoint *fragments* of the reference path F_r^i , $i = 1, \dots, n_r$, where each F_r^i is a subset of p_r , defining a path, and such that

$$p_r = \cup_{i=1}^{n_r} F_r^i.$$

The number of edges of each fragment is picked from a set of available lengths with a given probability distribution \mathbf{P}_r , an algorithm parameter. Similarly, we create n_d fragments F_d^i , of the deception routes, with an associated probability distribution \mathbf{P}_d for the fragment lengths, also an algorithm parameter.

Then, we associate each fragment F_r^i , for $i = 1, \dots, n_r$ to one agent in random fashion, following a uniform probability distribution. Note that each fragment F_r^i is associated to only one agent. We also associate each fragment F_d^i , for $i = 1, \dots, n_d$ to an agent with probability p_n , and with probability $1 - p_n$, each fragment F_d^i remains unassigned and will not be traversed by an agent. Let E_k denote the total set of fragments associated in this way to agent k , for $k = 1, \dots, N$.

Further, each vertex $v_i \in V_A$ is assigned uniformly to the available agents. Again, each vertex in V_A is assigned to exactly one agent. Let V_k denote the set of vertices assigned to agent k .

For each agent k , for $k = 1, \dots, N$, we create path p^k as follows. Order all elements of the sets E_k and V_k randomly with a uniform distribution, yielding sequence S_k . Then, in path $p^k = \{e_i^k\}$, for $i = 1, \dots, n_k$, edge e_1^k connects the source, v_s , to one of the vertices (picked randomly with a uniform distribution) of the first element in S_k in case the latter is an edge, and to the first element itself, in case that element is a vertex. We continue in this way, connecting all elements in S_k , uniformly picking the order in which edges are traversed.

Note that in this way, we guarantee that each edge in p_r is traversed by exactly one agent, and similarly, that all vertices in V_A are visited by exactly one agent.

C. Path Evaluation

Path generation is fast and cheap. Out of the many possible path sets produced, we intend to pick one set that is the “most deceptive”.

First, we impose an additional constraint on path sets that stems from the following reasoning, in part based on experience. If an observer reports the sighting of two agents traversing the same vertex, associated with a street crossing, within a relatively small time interval ΔT , then, this indicates a particular importance of that location. In

order to treat all vertices equally, we therefore impose a minimal time difference ΔT_m between consecutive visits to any vertex. This additional constraint further reduces the set of candidate path sets.

Finally, for each set of agent paths, we determine the “best” path set as follows. As mentioned in section III-A, we assume the opponent has sufficient observers to detect agent traversals at all edges $e \in E_m$. Let m_r denote the number of agent observation along the reference path. We create m_r pairs $(s_1, t_1), \dots, (s_{m_r}, t_{m_r})$, one for each observation. Here, the coordinate s represents the distance along p_r ($s = 0$ at vertex v_s). Hence, s_i is the location along the reference path where the i th observation took place. Correspondingly, t_i is the time at which the i th observation occurred. We then compute the sample correlation coefficient ρ_{st}^r for the pairs $(s_1, t_1), \dots, (s_{m_r}, t_{m_r})$ as

$$\rho_{st}^r = \frac{s_{st}}{\sqrt{\sum (s_i - \bar{s})^2} \sqrt{\sum (t_i - \bar{t})^2}}, \quad (1)$$

where

$$\begin{aligned} s_{st} &= \sum_{i=1}^{m_r} (s_i - \bar{s})(t_i - \bar{t}), \\ \bar{s} &= \frac{1}{m_r} \sum_{i=1}^{m_r} s_i, \\ \bar{t} &= \frac{1}{m_r} \sum_{i=1}^{m_r} t_i. \end{aligned}$$

The numerator in Eq. (1) is the covariance between two random variables s and t , while the denominator is the product of the standard deviations of s and t , respectively. The value ρ_{st}^r close to one, indicates a close to linear dependence between s and t along the reference path. In words, the agents traverse fragments of p_r almost in order, from start to target. For ρ_{st}^r close to -1 , we have that p_r is traversed in almost linearly in the opposite direction. Both cases are easy clues for the opponent, and therefore we intend to minimize $|\rho_{st}^r|$, but not only on the reference path, also on deception routes. In particular, we similarly compute ρ_{st}^d along deception routes and take a linear combination of $|\rho_{st}^r|$ and all $|\rho_{st}^d|$, weighted by the physical length of each path to obtain the “total correlation” ρ of the path set. The path set leading to the smallest ρ is picked.

IV. EXAMPLE AND DISCUSSION

In this section we will demonstrate typical path sets created by the algorithm and highlight the implementation into a team of rotorcraft UAVs. We will also address certain operational characteristics of the randomized algorithm.

We wish to demonstrate the application of this problem solution on the arbitrary environment in Fig. 5 scaled based on the dynamics of an average fixed wing SUAV [1, 15]. With an average mission battery of 80 minutes and a cruise airspeed of 13.5 mps, we choose to establish the length of the reference trajectory to 20 km to allow each agent sufficient range to cover the route non-sequentially.

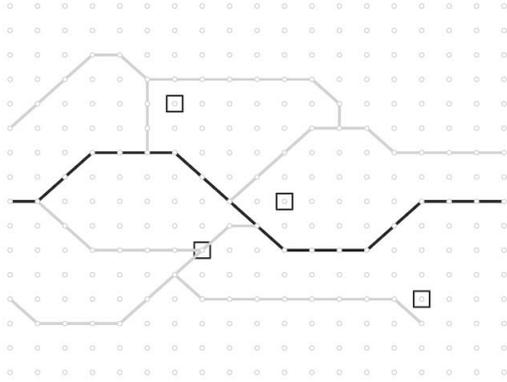


Fig. 5. Graph structure and notation. Vertices are circles, deception routes are light lines, the reference trajectory is the dark line and areas of interest are squares.

The battery life also establishes T_{max} as 80 minutes. The reference trajectory p_r , critical areas V_A , and deception routes E_p are apparent in Fig. 5. Finally we set the time required to remain at each critical urban area t_{v_i} as 120 seconds and the temporal spacing between agents, ΔT_m as 60 seconds. This completes the set of constraints required to generate and evaluate trajectories.

After performing limited preprocessing, we tested the speed and scalability of our approach constrained to 10 seconds of computation time to simulate real time applications. Using two agents, we selected a sample output for illustration. The trial created 665,811 path iterations satisfying the spatial constraints where 13,725 met T_{max} and 6279 further met ΔT_m . The trajectories of the two agents are depicted in Fig. 6.

Clearly, these agents not only satisfy the complete set of constraints, but they also represent an “optimal” mixed strategy. Given our objective to minimize $|\rho_{st}|$, the agents behave as expected. In general we find that the “optimal” trajectory for a given trial has multiple fragments along the reference trajectory, and the agents traverse these fragments in a random order. The inclusion of multiple deception routes also serves to minimize $|\rho_{st}|$ as the agents traverse these fragments in a similar manner. It is not possible to determine the reference trajectory, p_r even with complete information concerning the agents’ trajectories (*i.e.* $q = 1$). Fig. 7 illustrates the effect of this mixed strategy on the observers who do not have perfect information. For any given time window, the observers cannot establish with any degree of certainty the location of the reference trajectory, p_r . The final trial ρ_{st} of 0.19 indicates that the distance of the observations along the routes, s , and the relative time of the observations, t , were nearly independent.

One of the central goals of this research was not only to determine feasible path sets for teams of vehicles, but to do so in real time. To determine the success of this algorithm in such a real time application, we assembled a team of two autonomous helicopters each linked individually via TCP/IP protocol to a laptop ground station. One central ground station executed the algorithm and fed a set of

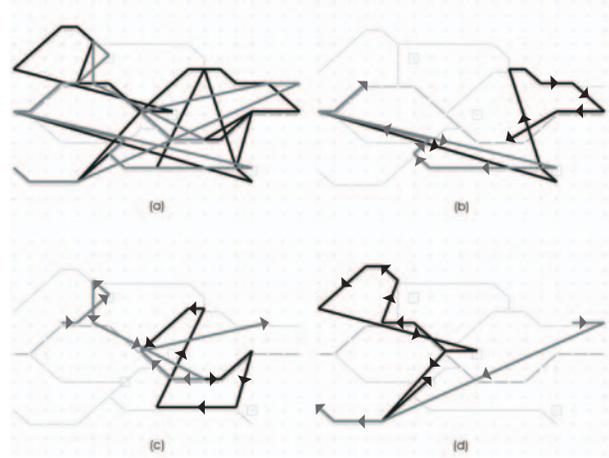


Fig. 6. (a) Complete trajectory for two agents. (b) Trajectory from origin to 30 minutes. (c) Trajectory from 30 minutes to 60 minutes. (d) Trajectory from 60 minutes to completion.

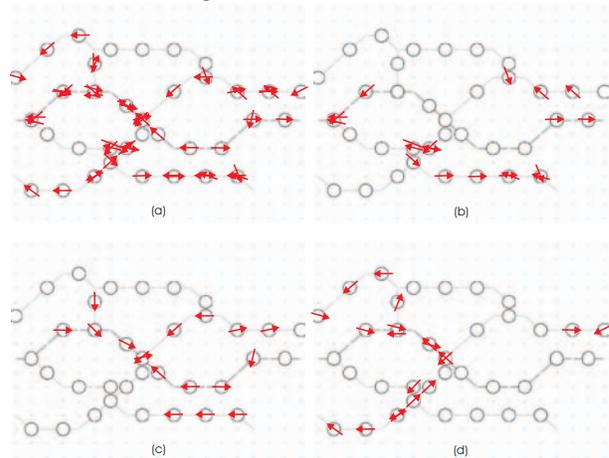


Fig. 7. (a) Complete observations from agents’ trajectories. (b) Observations from origin to 30 minutes. (c) Observations from 30 minutes to 60 minutes. (d) Observations from 60 minutes to completion

waypoints to each subordinate ground station who in turn relayed the coordinates to their respective aircraft. We chose to limit airspeed to 3.5 m/s for safety reasons as well as to allow the flight control logic to smoothly negotiate abrupt corners. Fig. 8 introduces the flight test environment, and Fig. 9 depicts the commanded waypoints output from the algorithm. The UAVs’ actual path, shown in Fig. 10 and 11, demonstrates that the guidance law follows the commanded trajectories closely while satisfying all mission requirements. This implementation supports our claim that the algorithm is well-suited to real time optimization tasks.

One potential drawback from using randomized algorithms in computing the “best” path sets is that we have no guarantee that the algorithm will converge for a given set of algorithm parameters. There are numerous factors that contribute to the convergence of the algorithm. For example, if p_n is unity for the case where the agents must traverse all edges in p_r and E_d , the constraint T_{max} must be correspondingly large for the algorithm to identify a feasible path set. To determine the relation between

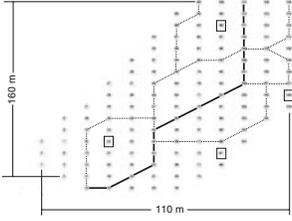


Fig. 8. Flight test environment. Dark line represents reference trajectory; light lines represent deception routes; squares represent areas of interest.

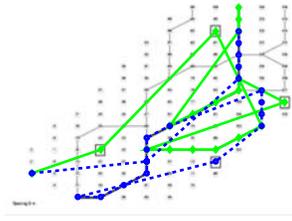


Fig. 9. Waypoints output from algorithm

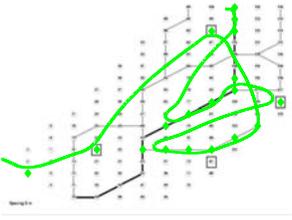


Fig. 10. Flight path of first vehicle

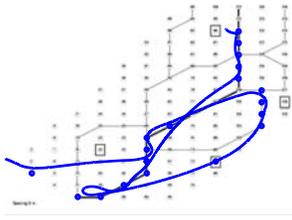


Fig. 11. Flight path of second vehicle

TABLE I
FACTORS AND THEIR EXPERIMENTAL RANGES

Factor	Min Value	Max Value	Step Size	Steps
p_n	0.0	1.0	0.2	6
T_{max}	2000 sec	6000 sec	500 sec	9
ΔT	60 sec	300 sec	60 sec	5
t_{v_i}	60 sec	300 sec	60 sec	5
T_c	0.5 sec, 1 sec, 2 sec, 5 sec, 10 sec			5

the algorithm parameters and convergence, we iterated the algorithm through all feasible combinations of parameters to determine if the algorithm would isolate a feasible path set within the given computational time, T_c (see Table I).

For each combination of factors we executed the algorithm ten times. This made for a total of $6 \times 9 \times 5 \times 5 \times 5 \times 10 = 249,750$ total algorithm executions. In general we found a strong correlation between T_{max} and algorithm convergence. If T_1 represents the time required for one vehicle to satisfy all mission requirements, we found that the algorithm never converges for $T_{max} \leq T_1$. Let T_2 represent T_1 scaled by the number of vehicles such that $T_2 = N \times T_1$. We found a maximum mission time, T_2 , whereby the algorithm converges for all combinations of parameters given that $T_2 \leq T_{max}$. The probability of assigning a deception route to an agent, p_n contributes significantly as T_{max} approaches T_2 . The expected loiter time at areas of interest t_{v_i} is highly correlated to convergence at T_{max} approaches T_1 . We did not find temporal spacing requirements, ΔT_m , to be a significant factor over the environments and number of agents that we tested. This statistical analysis provides a limited set of heuristics that identify major trends within the performance of the algorithm.

Another benefit of this approach is the ability to scale

the problem to any dimension without any appreciable alterations or performance downgrades. In the extreme, hundreds of agents could satisfy the set of constraints in a minimal amount of time while traversing every edge $e_{ij} \in E_m$ with uniform probability. The observations would be completely independent of each other, and the probability of the observers determining p_r would be minimal. Testing has not indicated any limits as to the number of agents feasible. The inclusion of additional agents tends to increase the percentage of iterations that satisfy T_{max} because each agent has a smaller set of fragments to traverse. An increase in the number of agents does not ease the ΔT constraint as the number of path intersections is independent of the number of agents.

V. CONCLUSION

We demonstrated that given sufficient agents and time available it is feasible for one player to render all information useless to the opponent. Even in cases with limited resources, a mixed strategy yields a trajectory set that deceives the opponent in establishing a threat to future movement. A randomized algorithm was presented that efficiently iterated through numerous feasible solutions to converge on a sufficiently rich solution. Implementation on a team of autonomous helicopters demonstrated the successful performance of the algorithm in real time applications.

REFERENCES

- [1] E. Bone and C. Bolkcom. Unmanned Aerial Vehicles: Background and issues for Congress. Report for Congress, The Library of Congress (Congressional Research Service), Apr 2003.
- [2] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, Cambridge, MA, second edition, 2001.
- [3] J. Devore. *Probability and Statistics for Engineering and the Sciences: Third Edition*. Brooks/Cole Publishing Company, Pacific Grove, CA, 1991.
- [4] S. Gentry and E. Feron. Modelling musically meaningful choreography. In *IEEE Systems, Man, Cybernetics Conference*, 2004.
- [5] M. Gondran and M. Minoux. *Graphs and Algorithms: Translated by Steven Vajda*. John Wiley and Sons, LTD, Chichester, NY, 1984.
- [6] Headquarters, Department of the Army. *FM 34-130: Intelligence Preparation of the Battlefield*, July 1994.
- [7] Headquarters, Department of the Army. *FM 101-5-1: Operational Terms and Graphics*, Sep 1997.
- [8] Headquarters, Department of the Army. *FM 1: The Army*, June 2001.
- [9] J. Hespanha, Y. Ateşkan, and H. Kizilocak. Deception in non-cooperative games with partial informatin. In *2nd DARPA-JFACC Symposium on Advances in Enterprise Control*, July 2004.
- [10] B. Jenkinson. Tactical observations from the Grozny combat experience. Master's thesis, US Army Command and General Staff College, 2002.
- [11] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, New York, NY, 2000.
- [12] B. Petit. Chechen use of the Internet in the Russo-Chechen conflict. Master's thesis, US Army Command and General Staff College, 2003.
- [13] W. Schneider, Jr. Unmanned Aerial Vehicles and Uninhabited Combat Aerial Vehicles. Study, Defense Science Board, Feb 2004.
- [14] D. Smith. Commonalities in Russian military operations in urban environments. Master's thesis, US Army Command and General Staff College, 2003.
- [15] D. Weatherington. Unmanned Aerial Vehicles roadmap: 2002 - 2027. Technical report, Office of the Secretary of Defense (Acquisition, Technology, and Logistics), Dec 2002.