Development of an Automated Tracking System of Tagged Wild Animals

Mariya Ishutkina[1] **Timothy Chan**[2] **Eric Feron**[3]

Abstract

The objective of this study is to develop a tracking algorithm which would allow a biologist to locate tagged animals using an unmanned aerial vehicle. The algorithm is developed to track the red wolves in a wildlife refuge in North Carolina. The red wolf is an endangered species and there are about a hundred of them living in the wild. For tracking purposes, each animal is outfitted with a collar containing a radio transmitter which emits a signal of a specific frequency over a given range. Using transmitter signal range, we discretize the refuge terrain map and develop a suboptimal *greedy* algorithm to search for the probabilistic targets striving to minimize the expected traveling time. To reduce the computational complexity, we consider a two-layered approach to the search problem. At the top level, the algorithm chooses the pack sectors as targets and, at the lower level, employs a local policy to travel between the discrete sensing locations. Using the animal behavior model provided by the wildlife biologists, we develop the probability rules to update the predicted target locations given the location of already found wolves. Finally, we present the results of a numerical simulation.

KEYWORDS: path planning, animal tracking, software, autonomous aircraft

Introduction

In 1967 red wolf (*canis rufus*) was listed as an endangered species and between years 1974 and 1980 the U.S. Fish and Wildlife Service captured a few remaining animals and established a captive-breeding program to restore red wolves in the wild. As pointed out on the Red Wolf Coalition website, the red wolves have been released back in the wild in the northeastern North Carolina since 1987. Currently, the wild populations of red wolves in the wildlife refuge in North Carolina number approximately a hundred animals. Each animal is outfitted with a collar containing a radio transmitter emitting a signal of a specific frequency. A biologist flies 2-3 times a week and manually scans for frequencies of various wolves which inhabit a certain territory. Since the wolves range over a large area, most of the telemetry is done from a fixed wing aircraft. A signal can be detected within a 3 km range at the altitude of 150 m. The total area of the wildlife refuge is 6,000 km^2. We propose to aid a biologist in tracking the animals by employing an unmanned aircraft equipped with an automated routine which guides the aircraft to predicted target locations.

In this paper, we describe the probability rules we use to predict the target locations based on known wolf dynamics. Red wolves live in extended family units called packs. A typical pack consists of five to eight animals which includes a breeding adult pair and offspring of different years. A typical pack usually roams over a home range of 40-200 km^2 depending on the habitat and prey density. According to the online edition of *The mammals of Texas*, the average home range of the pack is 56 km^2. Red wolves are territorial animals and a wolf pack would actively defend its home range against other wolves. Using the telemetry data analysis outlined by Millspaugh et al. (2001) and Williams et al. (2002) and the current searching patterns of a wildlife biologist, one is able to derive a model to predict the location of the animals. For example, one can predict that the animal's movement is usually constrained to its home range with the preference given to the locations of habitual hunting territory and its den. Since the red wolves live in packs, one is able to derive certain probability rules which would predict the location of other pack members if one of the pack wolves has been located. The full model of daily wolf behavior dynamics is currently under development by the biologists and it includes such factors as pack membership and home range, season, sex and social status of a wolf, geographic roam and a possible travel range for a wolf between the observation times.

The tracking problem can be formulated as a search problem where hidden targets are sought by an autonomous robot with local sensory information, where the wolves represent the targets and the autonomous robot is an unmanned aerial vehicle (UAV). Since the speed of the aircraft is far greater than the speed of the animals, the target positions, described by the probability density functions, are considered to be stationary. The robot surveys the terrain detecting and recording the targets' locations, updating the probabilities and performing a real-time trajectory planning while aiming to minimize the expected traveling time needed to discover all the targets. This search problem is essentially a Euclidean Traveling Salesman Problem (TSP) which strives to find a path of minimum Euclidean distance between points in a plane which includes each

[1]Ph.D. candidate, Department of Aero/Astro, MIT, Cambridge, MA 02139. Email: ishut@mit.edu.
[2]Ph.D. candidate, Operations Research Center, MIT, Cambridge, MA 02139. Email: tcychan@mit.edu.
[3]Associate Professor of Aeronautics and Astronautics, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, 77 Massachusetts Ave, Room 32-D724, Cambridge, MA 02139. Email: feron@mit.edu.

point exactly once and returns to its starting location. The TSP is an NP-complete problem which means that there is no known algorithm capable of finding the optimal path in polynomial time, however, there exist several heuristic tools capable of finding a numerical solution to the TSP. Due to the problem's complexity, most of the approaches which seek to find an optimal path based on tools such as dynamic programming, scale poorly with the number of cells and are not practical. Other approaches, such as proposed by DasGupta et al. (2004) and Sarmiento et al. (2003) produce heuristic based polynomial-time suboptimal solutions to the discrete search problem. Several authors, including Jin et al. (2003), Frazzoli and Bullo (2004) and Hespanha et al. (1999) have also considered a related dynamic vehicle routing problem for groups of autonomous vehicles seeking to minimize the expected waiting time to find stochastically-generated targets. In addition, recent work by Kanchanavally et al. (2004) aims to solve a prediction and search problem. The prediction aspect of the problem is captured through the use of hospitability maps which provide a likelihood for each point on the terrain surface. The search aspect is captured by a cooperative search algorithm which uses a non-gradient-based nonlinear programming method of surrogate optimization outlined by Torczon et al. (1998). Our work distinguishes itself by focusing on a real-life application of a search problem by developing and implementing a suboptimal, scalable and effective algorithm. We also develop the probability update rules which are able to predict the target locations based on the biological dynamic wolf model.

Problem Formulation

We grid the navigation terrain into a finite number of sectors whose size is based on an average geographic home range of a wolf pack. Then, we divide each sector using a square grid and represent each subsector by a node as shown in Figure 1. Each node is identified by a pair of coordinates (i, j). The subsector size is chosen such that when the aircraft is located at the node, it can scan the whole subsector at one unit of time. We denote the length of the subsector's edge as $2r_0$, where $r_0 = r_t/\sqrt{2}$ and r_t is the target's signal range. This choice of the subsector size ensures that certain points on the boundaries between subsectors overlap and are scanned twice as the aircraft moves between the neighboring sectors.

The aircraft is restricted to move only between the neighboring nodes. At the beginning of the search we assign *a priori* probability of finding wolf l at node (i, j) for each node of the grid. During the search, we consider the position of individual animals to be spatially invariant with respect to the position of the aircraft. Let ${}^l\Pi_k^{ij}$ be the probability that wolf l is at location (i, j) at the beginning of period k. As the aircraft moves to the search location (i, j), the probabilities ${}^l\Pi_k^{ij}$ for each wolf are updated. If a wolf is detected at (i, j), then the neighboring probabilities are updated using the probability update rules and a new searching target is selected using a greedy heuristic. The actual geographic grid is rather sparse due to the presence of natural boundaries, such as water barriers, which restrain the wolves' movement.

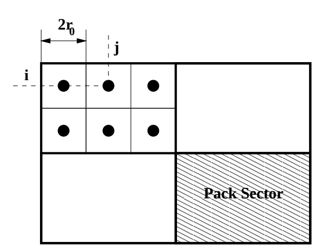

Figure 1: The mapping of the map onto a graph. The dots represent the nodes. At a node, the aircraft updates the probabilities of locating a wolf at neighboring nodes and chooses its next search target.

Probability Update Rules

We present the probability update rules for only three wolves. The formulation, however, can be easily extended to an arbitrary number of animals and packs. We assume that w_1 and w_2 are two wolves from the same pack, and w_3 is a lone wolf or a wolf from a different pack. Bertsekas (1995) formulates this problem as the *Search Problem* on pp. 247-248. The search at each node can have three outcomes:

- Search node (i, j) and don't find anything.
- Search node (i, j) and find either wolf w_1 or w_2.
- Search node (i, j) and find wolf w_3.

Scenario 1

After searching the site (i,j) at time k and not finding anything, the probability evolves as

$$ {}^{l}\Pi_{k+1}^{ij} = \frac{{}^{l}\Pi_{k}^{ij}(1-\beta)}{{}^{l}\Pi_{k}^{ij}(1-\beta) + 1 - {}^{l}\Pi_{k}^{ij}} \tag{1} $$

using Bayes' rule, where β be the probability of detecting the wolf at location (i,j) given that the wolf is present at that location. In practice, this value is close to 1. One can show that ${}^{l}\Pi_{k+1}^{ij} \leq {}^{l}\Pi_{k}^{ij}$, and in particular, if ${}^{l}\Pi_{0}^{ij} < 1$, then ${}^{l}\Pi_{k+1}^{ij} < {}^{l}\Pi_{k}^{ij}$ for all k.

Let $\mathcal{R}(i,j)$ be the set of points that a wolf can reach in a day if it was at (i,j) the previous day. This can be thought of as the one-day "range" of a wolf. Also, let ${}^{l}\mathcal{R}_d$ be the set of locations where wolf l can be with positive probability on day d. Note that if wolf l was found at location (i,j) on day d, then ${}^{l}\mathcal{R}_{d+1} = \mathcal{R}(i,j)$. If wolf l was not found, then ${}^{l}\mathcal{R}_{d+1} = \cup_{(y,z)\in {}^{l}\mathcal{R}_d} \mathcal{R}(y,z)$. With the notation $|\mathcal{A}|$ denoting the cardinality of the set $\mathcal{A}$, we can write

$$ {}^{l}\Pi_{k+1}^{yz} = {}^{l}\Pi_{k}^{yz} + \frac{{}^{l}\Pi_{k}^{ij} - {}^{l}\Pi_{k+1}^{ij}}{|{}^{l}\mathcal{R}_d| - 1} $$

for all $(y,z) \in \mathcal{R}(i,j)$ and $(y,z) \neq (i,j)$ if we are searching in day d. That is, the probability is distributed uniformly to those states that the wolf could be in during day d if we did not find him at some node (i,j).

Scenario 2

After searching the site (i,j) at time k, assume we found a pack wolf (e.g. wolf w_1), then the updated probabilities are

$$ {}^{w_1}\Pi_{k+1}^{ij} = 1 \text{ and } {}^{w_1}\Pi_{k+1}^{yz} = 0 \quad \forall (y,z) \neq (i,j). $$

We define $\mathcal{N}(i,j)$ to be the "neighborhood" of (i,j). It is a set of locations which is "close" to (i,j) and relates to how close pack wolves tend to stay together in their home range. ${}^{w_2}\Pi_{k+1}^{ij}$ and ${}^{w_3}\Pi_{k+1}^{ij}$ are updated first according to Eq. (1) and then the other probabilities are updated as

$$ {}^{w_2}\Pi_{k+1}^{yz} = \begin{cases} \frac{1}{\alpha}\,{}^{w_2}\Pi_{k}^{yz} & \text{if } (y,z) \notin \mathcal{N}(i,j) \cap {}^{w_2}\mathcal{R} \\ {}^{w_2}\Pi_{k}^{yz} + \dfrac{\sum\limits_{(y,z)\notin\mathcal{N}(i,j)} \left({}^{w_2}\Pi_{k}^{yz} - \frac{1}{\alpha}\,{}^{w_2}\Pi_{k}^{yz}\right) + {}^{w_2}\Pi_{k}^{ij} - {}^{w_2}\Pi_{k+1}^{ij}}{|\,\mathcal{N}(i,j) \cap {}^{w_2}\mathcal{R}\,| - 1} & \begin{array}{l}\text{if } (y,z) \neq (i,j) \\ \text{and } (y,z) \in \mathcal{N}(i,j) \cap {}^{w_2}\mathcal{R}\end{array} \end{cases} $$

and

$$ {}^{w_3}\Pi_{k+1}^{yz} = \begin{cases} \frac{1}{\gamma}\,{}^{w_3}\Pi_{k}^{yz} & \begin{array}{l}\text{if } (y,z) \neq (i,j) \\ \text{and } (y,z) \in \mathcal{N}(i,j) \cap {}^{w_3}\mathcal{R}\end{array} \\ {}^{w_3}\Pi_{k}^{yz} + \dfrac{\sum\limits_{\substack{(y,z)\in\mathcal{N}(i,j) \\ (y,z)\neq(i,j)}} \left({}^{w_3}\Pi_{k}^{yz} - \frac{1}{\gamma}\,{}^{w_3}\Pi_{k}^{yz}\right) + {}^{w_3}\Pi_{k}^{ij} - {}^{w_3}\Pi_{k+1}^{ij}}{|\,\mathcal{S} \setminus \mathcal{N}(i,j)\,|} & \text{if } (y,z) \notin \mathcal{N}(i,j) \cap {}^{w_3}\mathcal{R} \end{cases} $$

These equations basically state that the probability of finding a pack wolf outside of the neighborhood of its pack-mate is reduced by some factor $\alpha \geq 1$, and similarly, the probability of finding a pack wolf in the neighborhood of a lone wolf (or a different pack) is decreased by another factor $\gamma \geq 1$. Naturally, for each wolf l, there will be no change in the probabilities at states outside of ${}^{l}\mathcal{R}$. The actual implementation of the probability update rules also differentiates between the scenario when the found pack wolf is found on or outside of its regular home range.

Scenario 3

After searching the site (i,j) at time k, assume we found a lone wolf, w_3. Then the updated probabilities are

$$ {}^{w_3}\Pi_{k+1}^{ij} = 1 \text{ and } {}^{w_3}\Pi_{k+1}^{yz} = 0 \quad \forall (y,z) \neq (i,j), $$

and ${}^{w_1}\Pi_{k+1}^{ij}$ and ${}^{w_2}\Pi_{k+1}^{ij}$ are updated according to Eq. (1). The ${}^{w_1}\Pi_{k+1}^{yz}$ and ${}^{w_2}\Pi_{k+1}^{yz}$ where $(y,z) \notin (i,j)$ are updated similarly to the ${}^{w_3}\Pi_{k+1}^{yz}$ update in the Scenario 2 thereby reducing the probabilities of finding wolves from a different pack in the neighborhood of a lone wolf.

Likelihood Metrics

In addition to updating the probabilities of individual wolves according to the rules described above, at each time step k, we update the node and sector likelihood metrics. The node likelihood $Ln_k(i,j)$ is defined as the probability of finding at least one wolf at the node (i,j), i.e.

$$Ln_k(i,j) = 1 - \prod_{l=1}^{\text{total \# wolves}} \big(1 - {}^l\,\Pi_k^{ij}\big). \tag{2}$$

Analogously, we define the sector likelihood $Ls_k(M)$ as the probability of finding at least one wolf at the sector M, i.e.

$$Ls_k(M) = 1 - \prod_{l=1}^{\text{total \# wolves}} \big(1 - \sum_{\text{nodes } (i,j) \in \text{ M}} {}^l\Pi_k^{ij}\big). \tag{3}$$

After the flight is completed, the transitional probabilities are updated to predict *a priori* probability distributions for each wolf in the next day (${}^l\mathcal{R}_{d+1}$). The transitional probabilities are updated using a stochastic dynamic model of daily wolf behavior.

Search Algorithm

We now formulate the search problem as a Dynamic Program (DP). To simplify notation, let us assume that we are searching for only one target. We define V to be the value of finding the target and $\mathcal{D}(i,j) = \{(i,j)\} \cup \{\text{set of points that are 1 step away from}(i,j)\}$. Then, for every initial state (i,j), the optimal cost $J^*(i,j)$ of the problem is equal to $J_0(i,j)$, given by the last step of the following algorithm, which proceeds backward in time from period $N-1$ to period 0:

$$\begin{aligned} J_N\big((i,j),\Pi_N\big) &= -\text{cost of returning from } (i,j) \text{ to home base}, \\ J_k\big((i,j),\Pi_k\big) &= \max_{(y_k,z_k)\in\mathcal{D}(i,j)} \Big[-\text{cost of travel } + \Pi_k^{y_k z_k}\beta V + (1-\Pi_k^{y_k z_k}\beta) J_{k+1}\big((y_k,z_k),\Pi_{k+1}\big)\Big]. \end{aligned}$$

Then the optimal control policy is represented by the sequence $\pi^* = \{\mu(y_0^*, z_0^*), \ldots \mu(y_{N-1}^*, z_{N-1}^*)\}$. We define N as an upper bound on the number of nodes one can cover and

$$\begin{aligned} \text{cost of travel} &= \lambda_V * \text{ travel time } + \mu, \quad \text{where} \\ \lambda_V &= \text{a factor which relates travel time to value } (V) \text{ of finding the target}, \\ \mu &= \text{a fixed cost}, \\ \text{travel time} &= \frac{\text{distance}\big((i,j),(y,z)\big)}{\text{aircraft velocity}}. \end{aligned}$$

Similarly, for an appropriate factor λ_R

$$\text{cost of return } = \lambda_R * \frac{\text{distance}\big((i,j),\text{base}\big)}{\text{aircraft velocity}}.$$

Note that this DP formulation reduces to the Search Problem in Bertsekas (1995) if the *cost of travel* is constant and $J_N\big((i,j),\Pi_N\big) = 0$.

The presented DP formulation is illustrative and can be extended to more than one target, however, as the number of targets is increased, the exact solution to the problem becomes prohibitively expensive. As a result, we present a suboptimal search algorithm based on a heuristic greedy policy which is solved in polynomial time.

As described in the Problem Formulation section, the navigation terrain is divided into pack sectors, which are in turn subdivided into individual node subsectors. The greedy algorithm works on two levels as follows. First, it considers the pack sector grid with the sector likelihood metrics calculated according to Eq. (3). It picks the sector with the maximum likelihood as its destination and solves the shortest path problem from the aircraft's current position to the destination. The shortest path is solved for a connected graph where the vertices represent the centers of each sector and the edge cost between sectors S_i and S_j is defined as

$$\text{cost}(S_i,S_j) = \frac{Ls_k(S_j)}{\text{travel time}(S_i,S_j)}. \tag{4}$$

Once the shortest path has been chosen, the aircraft starts flying to the maximum likelihood sector following the path between sectors' centers. As the aircraft passes each node location, it takes measurements, and if a wolf is detected on the way to the maximum likelihood sector, the algorithm operation switches to the second level in the current sector. If no wolf is detected on the way to the destination, the aircraft reaches the maximum likelihood sector and then switches to the second level.

The second level consists of computing the shortest path on a graph where the vertices are the individual nodes within the sector. Here, the edge costs between the graph vertices are given by Eq. (4) except that the sector likelihood is replaced by the node likelihood Equation (2). The destination of the new shortest path is the maximum likelihood node of the current sector. The second level shortest path is recomputed every time a wolf is detected, otherwise, the aircraft follows the original shortest path. The algorithm switches back to the first level either when the time threshold is reached or the sector likelihood drops below a certain threshold. Once at the first level, the algorithm chooses the maximum likelihood sector as its destination and computes a shortest path between the sectors. The individual node probabilities are updated at each time step and the node and sector likelihoods are updated only when a wolf is found and/or a new destination needs to be chosen.

The breakdown of the search algorithm into two levels reduces the number of graph vertices, scales better and is flexible because of granularity. To find the shortest path in a weighted directed graph where all edge costs are nonnegative, we use Dijkstra's Algorithm described on pp. 527–531 of Cormen et al. (1990). The runtime of Dijkstra's Algorithm is $O(n^2)$.

Example

To illustrate the feasibility of our searching approach, we present the results of a numerical simulation. The actual navigation terrain is shown in Figure 2. It is roughly bounded by two airports: Manteo in the East, and Plymouth in the West. The terrain can be represented by a 30×40 node grid with about a hundred and thirty sectors each consisting of nine nodes. The grid structure is sparse due to the presence of uninhabited areas, therefore many sectors incur zero likelihood and do not need to be searched. Only about forty of these geographic sectors are inhabitable pack sectors. The complete behavior model of the whole wolf population is still under development by the biologists. Therefore, we present results of a realistic simulation for only a 9×12 node grid with four packs and a total of eight wolves using a preliminary wolf model. Figures 3– 6 show the results of the simulation at different time steps. The figures show the node likelihoods represented by the arrows and the aircraft path up to that time point; the stars represent the actual wolf locations. In this simulation the wolf located at $(6, 7)$ is a lone wolf who is outside of its home range with a widely dispersed range $\mathcal{R}$. As a consequence, the algorithm takes longer time to find it. The presented search algorithm works best if the probability density functions for each target are representative of their actual locations. On average, for the eight wolf search problem, the algorithm takes about three times longer to find all the targets than the optimal path computed for the deterministic targets.

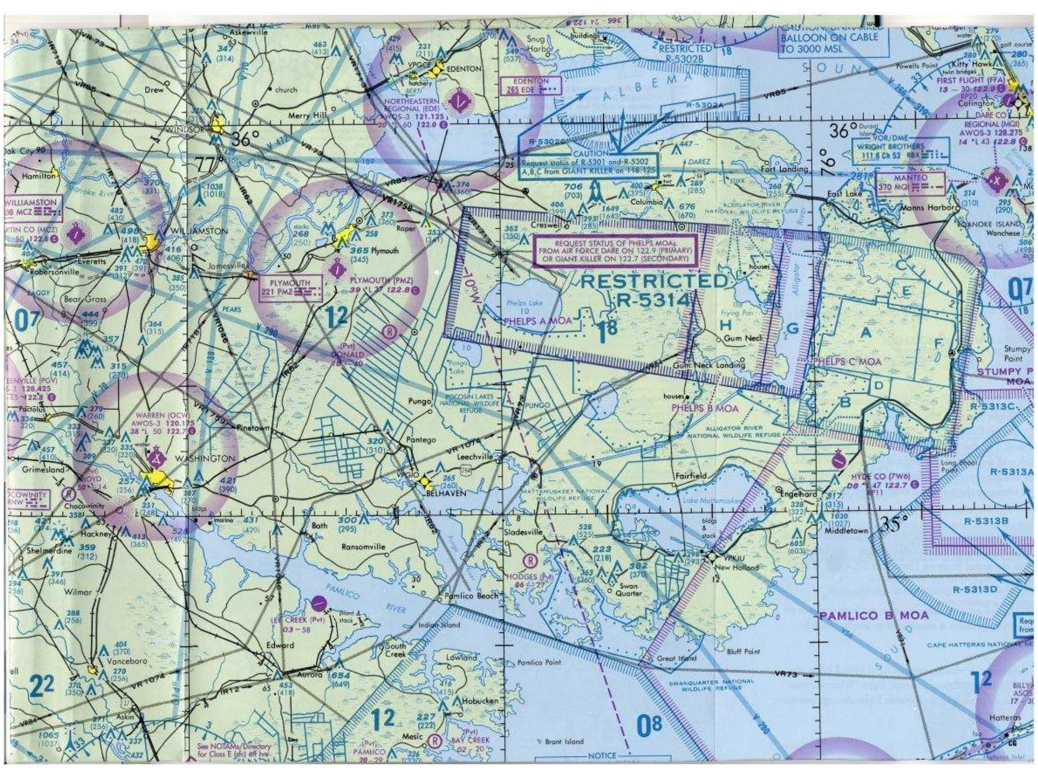

Figure 2: Map of the navigation terrain

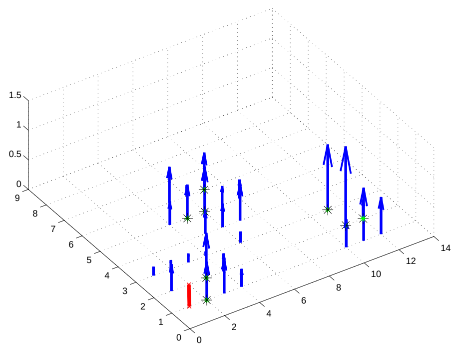

Figure 3: Time = 1

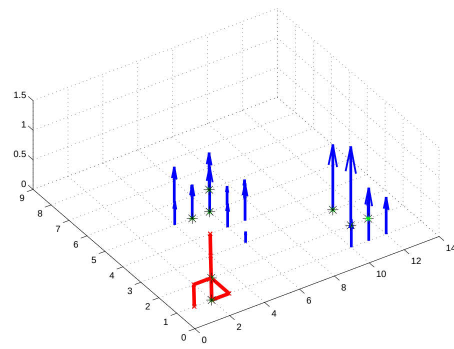

Figure 4: Time = 7

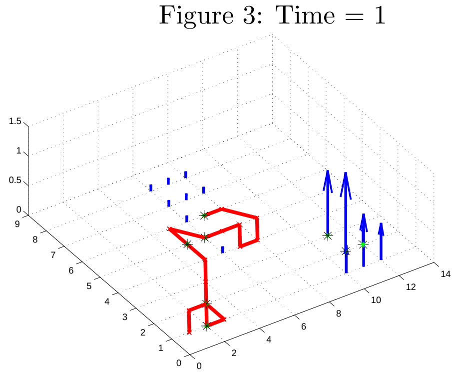

Figure 5: Time = 17

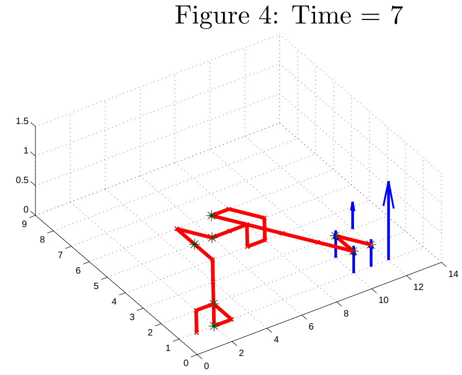

Figure 6: Time = 23

Conclusions and Future Work

In this paper we have described the search algorithm and probability update rules for finding the tagged wolves in the wildlife refuge. We consider the refuge terrain to be a discrete map with a finite number of sectors which are in turn comprised of nodes. We describe the search algorithm which allows for reduction of problem complexity by breaking the search task into two levels. At the first level, the algorithm searches for a path among geographic sectors and at the second level, the algorithm searches for a path among the individual nodes. We also present simulation results using a preliminary eight wolf model at a specific geographic location in the park.

The next step of this work is to implement the algorithm on the actual hardware. For demonstration, we are going to use a small-scale problem using people as the search targets and the X-Cell .60 helicopter as the UAV platform. In parallel, we are developing the complete grid with the appropriate probability distribution rules for the whole refuge based on the available historical behavioral data. Once the full model is incorporated, it will be tested on a manned fixed wing aircraft. The final goal of this research is to allow for autonomous tracking of the tagged wildlife using a UAV. We are currently considering two commercially developed long range (1,500 – 3000 km) UAV platforms: Insitu Seascan and Aerosonde.

To date, the UAVs have been used successfully for specialized weather and military surveillance activities. However, the Federal Aviation Administration does not currently have a policy for UAVs flying in the National Aerospace. According to the Wolf Source website, the current red wolf recovery area comprising 6000 km^2 includes three national wildlife refuges, a Department of Defense bombing range, other public and private lands (see Figure 2). Since the red wolf population for this project inhabits the rural area with a small human population, we hope to entice the FAA approval for the flight path in the National Airspace.

Acknowledgments

We would like to thank Michael Morse of the Red Wolf Recovery Program at the Alligator River National Wildlife Refuge.

References

1. Bertsekas, D. 1995. *Dynamic Programming and Optimal Control*. Athena Scientific, Belmont, MA.

2. Cormen, T.H., C. E. Leiserson, and R. L. Rivest. 1990. *Introduction to Algorithms*. MIT Press, Cambridge, MA.

3. DasGupta, B., J. Hespanha, and E. Sontag. 2004. Computational Complexity of Honey-pot Searching with Local Sensory Information. In *Proc. of the 2004 Amer. Contr. Conf.*, to appear.

4. Frazzoli, E., and F. Bullo. 2004. Decentralized algorithms for vehicle routing in a stochastic time-varying environment. In *Proc.* 43^{rd} *IEEE Conference on Decision and Control*, to appear.

5. Hespanha, J.P., H. J. Kim, and S. Sastry. 1999. Multiple-Agent Probabilistic Pursuit-Evasion Games. In *Proc. of the* 38^{th} *Conference on Decision and Control*, Vol 3, pp. 2432-2437.

6. Jin, Yan, A.A. Minai, and M.M. Polycarpou. 2003. Cooperative real-time search and task allocation in UAV teams. In *Proc.* 42^{nd} *IEEE Conference on Decision and Control*, Vol 1, pp. 7-12.

7. Kanchanavally, S., R. Ordóñez, and J. Layne. 2004. Mobile Target Tracking by Networked Uninhabited Autonomous Vehicles via Hospitability Maps. In *Proc. of the 2004 Amer. Contr. Conf.*, to appear.

8. Millspaugh, J., and Marzluff, J., Eds. 2001. *Radio Tracking and Animal Populations*. Academic Press, San Diego, CA.

9. Sarmiento, A., R. Murrieta, and S.A. Hutchinson. 2003. A strategy for searching an object with a mobile robot. In *Proc. Int. Conf. on Advanced Robotics 2003*.

10. Torczon, V., and M. Trosset. 1998. Using approximations to accelerate engineering design optimization. Tech. Rep. ICASE Report No. 98-33, NASA/CR, Sept. 1998.

11. Williams, B.K., J.D. Nichols, and M.J. Controy. 2002. *Analysis and Management of Animal Populations*. Academic Press, San Diego, CA.

12. The Red Wolf Coalition website. Available at http://www.redwolves.com. Accessed 2 June 2004.

13. The Wolf Source website. Available at http://www.wolfsource.org/redwolf.html/. Accessed 2 June 2004.

14. The mammals of Texas online edition. Available at http://www.nsrl.ttu.edu/tmot1/canirufu.htm. Accessed 2 June 2004.

15. The Aerosonde UAV website. Available at http://www.aerosonde.com/drawarticle/42. Accessed 3 July 2004.

16. The Insitu miniature robotic aircraft website. Available at http://www.insitugroup.net. Accessed 3 July 2004.