Multiagent Flight Control in Dynamic Environments with Cooperative Coevolutionary Algorithms

Mitchell Colby
Oregon State University
colbym@engr.oregonstate.edu

Matt Knudson
NASA Ames Research Center
matt.knudson@nasa.gov

Kagan Tumer
Oregon State University
kagan.tumer@oregonstate.edu

Abstract

Dynamic flight environments in which objectives and environmental features change with respect to time pose a difficult problem with regards to planning optimal flight paths. Path planning methods are typically computationally expensive, and are often difficult to implement in real time if system objectives are changed. This computational problem is compounded when multiple agents are present in the system, as the state and action space grows exponentially. In this work, we use cooperative coevolutionary algorithms in order to develop policies which control agent motion in a dynamic multiagent unmanned aerial system environment such that goals and perceptions change, while ensuring safety constraints are not violated. Rather than replanning new paths when the environment changes, we develop a policy which can map the new environmental features to a trajectory for the agent while ensuring safe and reliable operation, while providing 92% of the theoretically optimal performance.

Introduction

Mobile robot coverage tasks such as those involving Unmanned Aircraft Systems (UAS) are continuously becoming more prevalent in industrial, military, and academic applications, in part due to their fast deployment times and ability to reach areas that ground locomotive systems cannot reach (Caballero et al. 2008). One important area of research is payload directed flight, where UAS must obtain as much information from an area as possible in a given amount of time, and potentially change their flight plans based on dynamic information obtained from the environment (Lee, Yeh, and Ippolito 2010). Traditional search algorithms are capable of developing flight plans, but are often too computationally expensive for dynamic multiagent environments. The dynamic nature of the environment requires altering flight plans in real time, which is computationally intractable in a multiagent system with an exponentially large search and action space.

In order to address the challenge of and further the ability to dynamically adjust routes in a multiagent payload directed flight system, this research incorporates cooperative coevolutionary algorithms, an extension of evolutionary algorithms for multiagent systems (Fogel 1994). Control policies which change flight trajectories based on dynamic environmental data are learned, allowing for real time trajectory adjustments based on changes in the state space. Such control policies become necessary as search algorithms become too slow to operate in real-time. In these missions, flight safety is extremely important; safety requirements, such as minimum separation between aircraft, must always be met. In this work, we present a cooperative coevolutionary algorithm which produces UAS control policies which ensure that system safety constraints are not violated.

Domain and Approach

We model multiagent payload directed flight as follows. Locations of potential Points of Interest (POIs) are arranged in an m by m grid where each point is one unit of distance apart from adjacent points. At the beginning of the experiment, n agents are initialized at random locations in the domain. At each timestep, only a subset of the POIs are observable. Over the course of an episode, all POIs are active once; the goal of the agents in the system is to observe each POI while ensuring minimum separation between aircraft are maintained. The system evaluation function is the number of POIs observed during the episode, minus a penalty for each time the safety constraint is violated. We use a cooperative coevolutionary algorithm to find control policies for the agents in the system.

The cooperative coevolutionary algorithm used in this research is a modification of that found in (Colby and Tumer 2012), with two key modifications. First, we add a large penalty to the system evaluation function for any violations of safety constraints, in order to encourage agents to learn safe policies. Second, if the algorithm begins converging to a solution which does violate safety constraints, mutation rates are increased in order to guide the algorithm to another region of the solution space.

Results

The potential POI locations were distributed in a 10 by 10 grid, with 10 agents moving within the environment. Each event in the simulation lasted for 25 timesteps, and 25 different POIs were active at any given time step. A comparison of

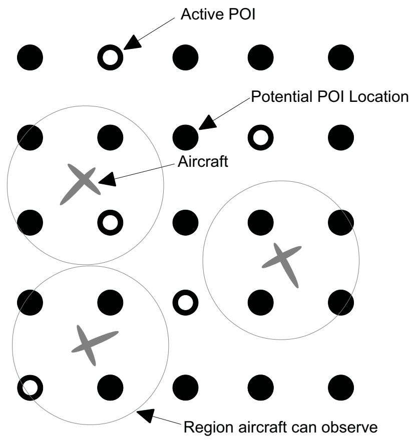

Figure 1: Payload Directed Flight domain. At any given moment, only a subset of the POIs are available to be observed. Aircraft have a finite region around them in which observations can be made.

our cooperative coevolutionary algorithm and a finite-time horizon recursive best first search is shown in Figure 2.

Due to the increased complexity of this domain resulting from the dynamic environment and increased number of POIs, a full search cannot be completed to create flight plans for an entire simulation. As the active POI locations change with respect to time, flight plans need to be dynamically changed in order to ensure that newly activated POIs are observed. Thus, we use a recursive best first search algorithm with a finite time window, where new flight plans were generated every 5 time steps. Every 5 time steps, an RBFS algorithm is completed to maximize the number of POIs to be observed over that time window, while ensuring that no safety violations occur. Although the flight plans are always optimal for the finite time window, they do not form

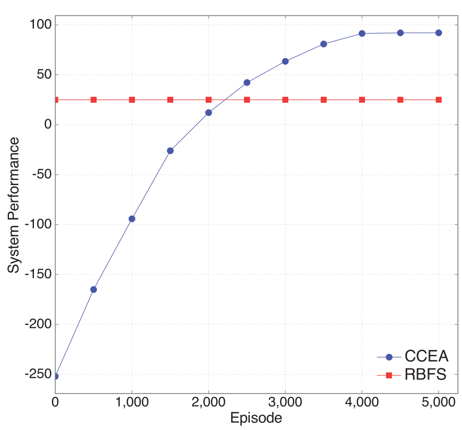

Figure 2: 10 by 10 grid with 10 agents. Finite time horizon RBFS results in 87% coverage, while the CCEA results in 92% coverage with no separation violations.

a globally optimal solution for the length of the entire simulation. This is seen in Figure 2, where the RBFS obtained an average coverage of 87% of the POIs in the domain.

For the CCEA, coevolution was allowed to proceed for 5000 generations. For mutation, 5 weights were selected at random from each network, and a random variable drawn from a Gaussian distribution with zero mean and a variance of 2 was added to each weight. There were 125 statistical runs conducted, with the error bars in Figure 2 reporting the error in the mean. As in Figure 2, the error bars are very small, and are often obscured by the plot symbols. After 125 statistical runs, the average performance of the CCEA corresponded to $92.05 \pm 1.56\%$ POI coverage, with a maximum of 100% coverage and a minimum of 88% coverage. Every single statistical run of the CCEA outperformed the average RBFS performance. There were no safety violations in any of the converged policies from any of the 125 statistical runs. As seen in Figure 2, the CCEA outperforms the RBFS search while ensuring that safety violations do not occur.

Discussion

As UAS tasks grow in complexity or the time available for these tasks decreases, adding more aircraft to the system allows for efficient completion of these tasks. However, adding more agents to the system results in an exponential growth of the state and action space, rendering traditional search algorithms intractable. The search algorithms may still be used with finite time horizons, but this results in severely suboptimal policies. In this research, we present a cooperative coevolutionary algorithm to develop control policies in multiagent payload directed flight. Our algorithm results in better learned performance than the finite time horizon search algorithms, while ensuring that safety constraints are still satisfied. The key contribution of this work is to demonstrate that multiagent learning can provide superior performance to traditional learning algorithms, while ensuring system constraints are not violated by the learned control policies.

Acknowledgements

This work was completed at NASA AMES Research Center.

References

Caballero, F.; Merino, L.; Ferruz, J.; and Ollero, A. 2008. Vision-Based Odometry and SLAM for Medium and High Altitude Flying UAVs. *Journal of Intelligent Robotic Systems* 54(1-3):137–161.

Colby, M., and Tumer, K. 2012. Shaping fitness functions for coevolving cooperative multiagent systems. In *Proceedings of the 11thInternational Joint Conference on Autonomous Agents and Multiagent Systems*, 425–432.

Fogel, D. 1994. An Introduction to Simulated Evolutionary Optimization. *Neural Networks, IEEE Transactions on* 5(1):3 –14.

Lee, R.; Yeh, Y.; and Ippolito, C. 2010. Geological Applications of Payload Directed Flight. *AIAA Infotech, Aerospace*.