Decentralized Cooperative Search in UAV's Using Opportunistic Learning

Yanli Yang, Ali A. Minai, Marios M. Polycarpou
ECECS Department
University of Cincinnati
Cincinnati, OH 45221-0030

Abstract

This paper addresses the problem of cooperative search in a given environment by a team of Unmanned Aerial Vehicles (UAVs). We present a decentralized strategy for cooperative search using an opportunistic cooperative learning (OCL) method. Our approach for cooperation is based on letting each UAV take into account the possible actions of other UAVs, such that the overall information about the environment is increased as rapidly as possible. Agents use adaptive predictors for this purpose, and share their predictors opportunistically to increase the overall performance of the team. The simulation results illustrate the proposed strategy.

1 Introduction

Cooperative control of Unmanned Aerial Vehicles (UAVs) has been an application of particular interest over the last few years. This paper focuses on the development and analysis of a decentralized strategy for cooperative search among a team of UAVs. The search task involves covering a region, locating and identifying entities and situations of interest, e.g. targets and threats. The basic idea is for a team of N vehicles to explore the environment rapidly, effectively, reliably and completely. Each UAV acts independently, planning its path based on its local and limited global information which comes from its own direct sensing and from other UAVs by means of communication. The path is planned dynamically in real time. While vehicles can certainly search the environment without cooperation (e.g., by following a pre-specified sweep pattern), the search can be made more efficient by using cooperation to minimize duplicated effort.

In previous work, we have described a recursive approach using multi-objective functions for the on-line cooperative search problem in [1, 2]. More recently, we have developed a simpler framework for the search problem using a discretized cellular space [3]. In both frameworks, UAVs move synchronously with a constant speed and are subject to restrictions on communication and maneuverability. In order to search the environment efficiently, each UAV needs to predict the state of its search neighborhood in the near future. This is done using feed-forward neural network predictors trained by a reinforcement learning(RL) algorithm [3]. Such algorithms have been used successfully in robotics and multi-agent systems [4]. In [3], we compared the search performance for the *centralized learning(CL)*, where all UAVs use a single, centrally trained adaptive predictor, with that of the *decentralized learning (DL)* case, where each UAV has its own predictor. Although the centralized approach provided better performance, this came at the cost of efficiency and robustness. To obtain the benefits of both approaches, we proposed the *opportunistic cooperative learning (OCL)* approach. In the OCL scenario, whenever two UAVs are in close proximity, they compare their predictors' performance, and the less successful UAV replaces its predictor by copying that of the other UAV with some probability. In this way, successful predictors tend to proliferate through the UAV population via a selection mechanism, without imposing rigidity of a single centralized predictor. Using simulations, we have shown that the OCL approach can provide prediction performance close to that of CL while retaining the advantages of DL [3].

In the present paper, we extend the previous work in several ways:

- The quality of search in the original formulation was based on the coverage of the environment.

In the current formulation, we include targets, and measure the quality of search in terms of the efficiency in finding targets.

- In the previous formulation, we used an arbitrary "certainty" variable that represented the degree to which a location had been explored. In the current approach, we use an approach suggested in [5] to obtain a formally defined measure of information updated by a Bayesian rule that accounts for sensor error.

- The previous work used "myopic" agents that only considered the reward obtainable at the step for which they were making a decision. In the present work, we also considered the prospects for future rewards.

The remainder of the paper is organized as follows. Section 2 briefly reviews some related research work. Section 3 presents the models for the UAVs and environment. Details of the decentralized cooperative search strategy are developed in Section 4 and the opportunistic learning algorithm is presented in section 5. Some simulation results are described in Section 6, which also includes a discussion of the results. Section 7 concludes the paper with some final observations. Throughout the manuscript, we use the terms "agent" and "UAV" interchangeably.

UAV cooperative control problems include target assignment [6], cooperative classification [7], and cooperative UAV rendezvous [8, 7]. Cooperative path planning is typically a part of all these problems, since they all involves timing or sequencing of UAVs for coordinated arrival at targets or other specified locations. The path planning required in these cases, however, is to find feasible paths from the UAV's initial position to its desired destination. Path planning during cooperative search, in contrast, is not entirely destination-oriented. In order to completely search a region, the vehicles can consider all areas of high uncertainty as their goal. This may occur in conjunction with the imperative to home in on known targets, as we discuss below.

The UAV cooperative search problem is also related to problems of multi-robot mapping and exploration, which is currently an active research topic in robotics [9, 10, 11, 12]. Some other related ideas and methods can also be found in the area of optimal search theory [13, 14].

2 Problem Definition

We consider a team of UAVs deployed in searching for targets in an environment of known size. The mission of the UAVs is to plan their trajectories so as to obtain the maximum amount of information about the environment in the shortest time, and to detect targets as rapidly as possible.

2.1 The Environment

The search *environment*, E, is represented as a $L_x \times L_y$ grid with periodic boundary conditions; each grid position is termed a *cell*. The environment is populated by stationary targets. The number and locations of the targets are initially unknown, and it is assumed that there is at most one target in each cell. The *state* of each cell, (x, y), is given by $s_{x,y} \in \{0, 1\}$, where $s_{x,y} = 1$ indicates that a target is present at (x, y), and $s_{x,y} = 0$ means that there is no target there. UAVs move in the environment in discrete time-steps, taking a sensor reading at each step (see below).

The UAV team works from a continuously updated *target probability map (TPM)*, $z_{x,y}(t) \equiv P(target\ present\ at\ (x, y))$. We term $z_{x,y}(t) \in [0, 1]$ the *target probability* of (x, y). The target probabilities for cells change as the UAVs scan them.

We also define the *uncertainty* associated with cell (x, y) as the Shannon entropy:

$$\begin{aligned} u_{x,y}(t) &= H[z_{x,y}(t)] \\ &= -z_{x,y}(t) log_2 z_{x,y}(t) \\ &\quad -(1 - z_{x,y}(t)) log_2 (1 - z_{x,y}(t)) \end{aligned} \tag{1}$$

Thus, if a cell (x, y) has $z_{x,y} = 0.5$, it has 1 bit of uncertainty, indicating complete ignorance by the UAVs of whether a target is present in that cell. Cells with $z_{x,y} = 1$ or 0 have an uncertainty of 0. We refer to the map of $u_{x,y}(t)$ as the *uncertainty map (UM)*.

Finally, we also use a binary variable $\zeta_{x,y}(t)$ to indicate whether a target has been confirmed or not in cell (x, y). Initially, all cells except those with known targets have $\zeta_{x,y} = 0$. The condition for updating is:

$$\zeta_{x,y}(t) = \begin{cases} 1 & if\ z_{x,y}(t) \geq \theta \\ 0 & else \end{cases} \tag{2}$$

where θ is a pre-defined threshold close to 1.

The TPM (and thus the UM) is initialized to reflect the *a piori* knowledge about possible target locations,

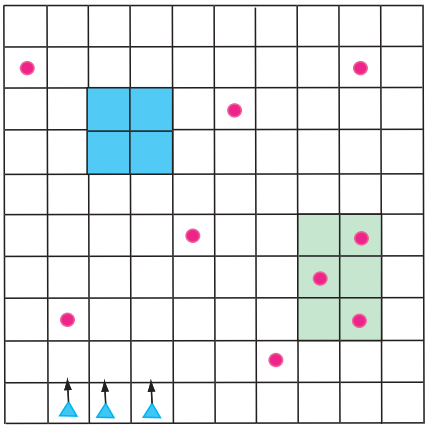

Figure 1: Example Search Environment: Triangles indicate UAVs and circles are targets. The darker gray region may denote a lake with a zero target probability and the lighter gray region may indicate a camp with a high target probability.

and is updated thereafter as UAVs take sensor readings. The initial target probabilities may reflect information such as known topographical features (e.g., lakes) where targets could not exist. Figure 1 illustrates an example TPM. We assume that all UAVs have access to the TPM, and, therefore, to the current uncertainty values for every cell.

2.2 The Agent Dynamics Model

The team consists of N identical UAVs moving synchronously in discrete-time, searching the environment for targets. Each UAV is equipped with a sensor (possibly multi-modality) and communication capabilities. Each UAV can be considered to be a point which can, at one step, move from the center point of one cell to the center point of another neighboring cell with maneuverability constraints and sense the new cell for a target using a sensor. We are assuming that the UAVs can communicate with other UAVs within their vicinity, and with the centralized TPM, within each time step.

The dynamics of the UAVs and their decision process is as follows. At time t, UAV i has cell position $(x_i(t), y_i(t))$, and can be in one of eight possible orientations, $o_i(t)$: 0 (north), 1 (northeast), 2 (east), 3 (southeast), 4 (south), 5 (southwest), 6 (west), and 7 (northwest). Each UAV plans its path q steps ahead of its current location, adding a new move at each time-step [1]. For this paper, we use $q = 2$. Thus, at time-step t, the UAV selects its position for $t+2$, the

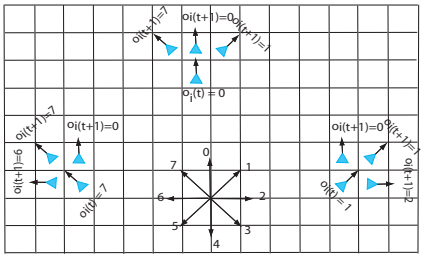

Figure 2: Possible move choices for UAVs in some orientations. Triangles indicate UAVs and arrows are the orientations.

position for $t+1$ having already been selected at step $t-1$. At time-step t, the UAV executes an *action* comprising the following three steps:

1. It chooses a new orientation, $o_i(t+2) \in \{o_i(t+1)-1, o_i(t+1), o_i(t+1)+1\}$ mod 8, i.e., the new orientation can change by at most one step. Note that $o_i(t+1)$ is known from step $t-1$.
2. It then designates the neighbor of $(x_i(t+1), y_i(t+1))$ facing orientation $o_i(t+2)$ as $(x_i(t+2), y_i(t+2))$.
3. Finally, it moves to grid location $(x_i(t+1), y_i(t+1))$ with orientation $o_i(t+1)$.

This essentially means that, at every step, the UAV either continues to move in the same direction as before or changes course to left or right by 45°, giving it three possible choices for step $t+2$. We designate these by $l_i = (x_i^l, y_i^l)$ (left), $f_i = (x_i^f, y_i^f)$ (front), and $r_i = (x_i^r, y_i^r)$ (right). Figure 2 shows this graphically for various orientations. Note that l_i, f_i, and r_i depend on time t, but we omit this in the notation for clarity. Limiting the UAVs' turning capability to 45° is a simplified way to reflect the curvature radius constraints usually applicable to realistic vehicles.

2.3 Bayesian Map Update Rule

The TPM is updated by incorporating sensor readings taken in each cell during the search. Because of the inevitable uncertainty in sensors' measurement and the intrinsic uncertainties of the terrain, the map values are updated probabilistically. Following sensor fusion practice in robotics navigation and obstacle avoidance [15, 5], we develop a Bayesian update rule (Equations (3) and (4)) for target probabilities. A visit by one UAV to cell (x, y) at time t update the cell's target probability value as follows:

• When the UAV's sensor reports a target present in the cell,

$$z_{x,y}(t) = \frac{p_s z_{x,y}(t-1)}{p_s z_{x,y}(t-1) + (1-p_s)(1-z_{x,y}(t-1))} \equiv F_1(z_{x,y}(t-1)) \quad (3)$$

• When this UAV's sensor reports there is no target in the cell,

$$z_{x,y}(t) = \frac{(1-p_s) z_{x,y}(t-1)}{p_s(1-z_{x,y}(t-1)) + (1-p_s) z_{x,y}(t-1)} \equiv F_0(z_{x,y}(t-1)) \quad (4)$$

Where p_s is the sensor fidelity. If several UAVs visit a cell simultaneously, the update is obtained by iterating equation (3) and (4).

To obtain (3) and (4), consider the case where a UAV takes a measurement in cell (x, y) at time t. Define the following for a cell (x, y):

- A is the event that $s_{x,y} = 1$, i.e., a target is located in cell (x, y).
- b_t is the binary sensor reading taken by the UAV, where $b_t = 1$ indicates target detection and $b_t = 0$ non-detection.
- B_{t-1} is the vector of all sensor readings for cell (x, y) by all UAVs taken upto time $t-1$ (i.e., before time t).
- $P(A)$ is the *prior probability* of a target being present in cell (x, y).

Based on the above definitions, $P(A \mid B_{t-1})$ is the probability of target existence in cell (x, y) at time $t-1$ and $P(A \mid B_{t-1}, b_t)$ is the updated probability after obtaining the new reading, b_t. Thus we have

$$z_{x,y}(t-1) = P(A \mid B_{t-1}) \quad (5)$$

$$z_{x,y}(t) = P(A \mid B_{t-1},\ b_t) \quad (6)$$

and $P(A \mid b_t)$ is the probability of the target existence given by the single sensor reading at time t. If the sensor has perfect discrimination, then $P(A \mid b_t = 1) = 1$ and $P(A \mid b_t = 0) = 0$. However, due to the uncertainties of sensor and the varieties of the terrains, there will be some noise in sensor readings so that $P(A \mid b_t)$ will be between 0 and 1 rather than be exactly 0 or 1.

To make the situation simple, we define the *fidelity* of the sensor's reading, denoted by p_s, as the probability of a correct sensor reading:

$$P(A \mid b_t = 1) = p_s \quad (7)$$

$$P(\bar{A} \mid b_t = 0) = p_s \quad (8)$$

$$P(A \mid b_t = 0) = 1 - P(\overline{A} \mid b_t = 0) = 1 - p_s \quad (9)$$

In the general case, the fidelity for the $b_t = 1$ and 0 cases need not be equal; we assume this for simplicity. Different kinds of sensors can have different fidelity, but here we assume that all UAVs have identical sensors. We also do not account explicitly for the geographically variable effects of terrain, but these could be built into our equations by making p_s a function of position in the environment.

We assume that the sensors' measurements in any cell are conditionally independent given the state of the cell, *i.e.*

$$P(b_1, b_2, \ldots, b_n \mid A) = \prod_{i=1}^{n} P(b_i \mid A) \quad (10)$$

Based on the above definitions and assumptions, the updating functions (3) and (4) follow directly from Bayes' rule using a method given by [5]. According to Bayes' rule,

$$\frac{P(A \mid B_{t-1}, b_t)}{P(\overline{A} \mid B_{t-1}, b_t)} = \frac{P(b_t \mid A, B_{t-1})}{P(b_t \mid \overline{A}, B_{t-1})} \cdot \frac{P(A \mid B_{t-1})}{P(\overline{A} \mid B_{t-1})}$$

which can be simplified by virtue of the conditional independence assumption to:

$$\begin{aligned} \frac{P(A \mid B_{t-1}, b_t)}{P(\overline{A} \mid B_{t-1}, b_t)} &= \frac{P(b_t \mid A)}{P(b_t \mid \overline{A})} \cdot \frac{P(A \mid B_{t-1})}{P(\overline{A} \mid B_{t-1})} \\ &= \frac{P(A \mid b_t)}{P(\overline{A} \mid b_t)} \cdot \frac{P(\overline{A})}{P(A)} \cdot \frac{P(A \mid B_{t-1})}{P(\overline{A} \mid B_{t-1})} \end{aligned} \quad (11)$$

By solving (11) for $P(A \mid B_{t-1}, b_t)$ using the fact that $P(\overline{A} \mid B_{t-1}, b_t) = 1 - P(A \mid B_{t-1}, b_t)$, we get

$$P(A \mid B_{t-1}, b_t) = 1 - \left[1 + \frac{P(A \mid b_t)}{1 - P(A \mid b_t)} \cdot \frac{P(\overline{A})}{P(A)} \cdot \frac{P(A \mid B_{t-1})}{P(\overline{A} \mid B_{t-1})}\right]^{-1} \quad (12)$$

Using equation (12), (5), (6), (7), (8), (9) and $P(A) = 0.5$, we can obtain the update equation (3) and (4) by exchanging $P(A \mid B_{t-1})$, $P(A \mid B_{t-1}, b_t)$, $P(A \mid b_t)$ with $z_{x,y}(t)$, $z_{x,y}(t+1)$ and p_s (or $1 - p_s$) correspondingly. A slightly more complicated expression results if $P(A) \neq 0.5$; the use of 0.5 is justified

by the UAVs' ignorance about the total number of targets in the system.

From update equation (3) and (4), we get the following for the change in target probabilities:

- When $b_t = 1$,

$$z_{x,y}(t+1) - z_{x,y}(t) = \frac{(2p_s - 1)(1 - z_{x,y}(t))}{p_s z_{x,y}(t) + (1 - p_s)(1 - z_{x,y}(t))} z_{x,y}(t) \quad (13)$$

- When $b_t = 0$,

$$z_{x,y}(t+1) - z_{x,y}(t) = \frac{(1 - 2p_s)(1 - z_{x,y}(t))}{(p_s(1 - z_{x,y}(t)) + (1 - p_s) z_{x,y}(t)} z_{x,y}(t) \quad (14)$$

Equation (13) and (14) show the relationship between the update rule and p_s for the $P(A) = 0.5$ case. When $p_s > 0.5$ (which means the sensor reading can give some valuable information), $z_{x,y}(t+1) > z_{x,y}(t)$ for $b_t = 1$ and $z_{x,y}(t+1) < z_{x,y}(t)$ for $b_t = 0$. When $p_s = 0.5$, the sensor readings are random, and $z_{x,y}(t+1) = z_{x,y}(t)$ in both (3) and (4), reflecting the fact that sensor readings provide no information about targets. Throughout this paper, we assume $p_s > 0.5$, i.e., the sensors are useful. In the general case, the sensors must be consistently informative, i.e., $P(A \mid b_t = 1) > P(A)$ and $P(\bar{A} \mid b_t = 0) > P(\bar{A})$, in order to provide useful estimates.

Note that, while we use the simplest version, the above formulation can easily incorporate features such as heterogeneous sensors, terrain-dependence, change in sensor characteristics, changing environmental conditions (e.g., light), etc., insofar as they are manifested in sensor fidelity.

2.4 Reward Definition

The goal of the UAVs' cooperative search can be stated in terms of two objectives:

1. **Coverage:** To reduce uncertainty over the whole environment as rapidly as possible.
2. **Target Detection:** To locate and verify as many of the existing targets as possible.

Since targets are likely to be relatively sparse in the environment, these objectives are not always mutually compatible. The need to detect and verify targets takes the UAVs towards regions with high target likelihood — i.e., cells with high z values — while the need for coverage requires that they go all over the environment. These two imperatives can be seen as the classic exploration vs. exploitation tradeoff in game theory and reinforcement learning [16].

In our formulation, each UAV makes its decision on which cell to visit at step $t+2$ by considering the rewards available for each of the three choices. The rewards are defined with respect to the two objectives as follows.

2.4.1 Coverage Reward for Exploring the Environment

The first objective is to reduce the total uncertainty about the environment as rapidly as possible. As discussed earlier, the uncertainty is simply the entropy of the target probability The reward, ρ_c, a UAV gets for sensing cell (x, y) and time t is the *change in uncertainty caused by the measurement's result*:

Based on the above definition, the *exploration reward* for searching cell (x, y) at time t, denoted as $\rho_e(t)$, can be defined as the *information entropy* of that cell at time $t-1$;

$$\rho_c(t) \quad = \quad u_{x,y}(t-1) - u_{x,y}(t) \quad (15)$$

Note that the change in entropy depends on the change in $z_{x,y}$, as given by (13) and (14). If more than one UAV visits cell (x, y) at the same time, the total change in uncertainty is divided equally among all visiting UAVs. Thus, it is better for UAVs to diversify their search paths from the exploration point of view, and the coverage objective is a *cooperative objective*. In seeking to achieve this, each UAV tries to take account of other UAVs in its vicinity and to estimate the effect of their actions on its own reward — and thus on the system objective of rapidly reducing uncertainty.

2.4.2 Reward for Finding Targets

The second objective is to maximize the number of targets found. Here, we must differentiate between targets that have already been verified sufficiently (i.e., they are "completely found") and those that are not. Only the latter should produce a reward

upon being found. We define the reward for finding a target by a single UAV at time t as:

$$\rho_f(t) = b_t(1-\zeta_{x,y}(t-1)) \quad (16)$$

where b_t is the measurement made by the UAV in cell (x,y) at time t. Unlike the case of the coverage reward, the target reward is specific to each UAV visiting (x,y), and is not shared with others visiting at the same time. Each UAV is rewarded based on its own sensor reading. Thus, this objective requires no cooperation.

2.4.3 Total Reward

The total reward $\rho(t)$ for one UAV visiting cell (x,y) at time t is defined as a linear combination of the above two kinds of rewards,

$$\begin{aligned}\rho(t) &= \alpha\rho_c(t) + (1-\alpha)\rho_f(t) \\ &= \alpha[u_{x,y}(t-1) - u_{x,y}(t)] \\ &\quad +(1-\alpha)b_t[1-\zeta_{x,y}(t-1)] \end{aligned} \quad (17)$$

where $\alpha \in [0,1]$. By changing α, the relative importance of the two objectives can be scaled. Note that when only one UAV enters (x,y) at time t, its reward depnds only on the $u_{x,y}$ values at times t (after update) and $t-1$ (before update), and on the UAV's sensor measurement.

When multiple UAVs enter cell (x,y) at time t, the coverage reward shared by all entering UAVs. Thus, if m UAVs enter cell (x,y) at time t, each UAV i gets reward

$$\begin{aligned}\rho(t) &= \frac{\alpha}{m}[u_{x,y}(t-1) - u_{x,y}(t)] \\ &\quad +(1-\alpha)b_t^i[1-\zeta_{x,y}(t-1)] \end{aligned} \quad (18)$$

where b_t^i indicates the measurement made specifically by UAV i. The reward for each UAV is, therefore, also a function of how many UAVs visit the cell.

3 Path Planning Method

The main task for each UAV at time-step t is to choose one of three moves for $t+2$ given the already fixed move for $t+1$. This is done by estimating (predicting) the expected reward for each 2-step target cell and selecting the one with the best payoff.

The reward estimate, $\hat{R}^i(k,t+2)$, for UAV i consists of two components:

Immediate Reward $\hat{\rho}^i(k,t+2)$: This is the reward the UAV expects upon entering a cell k at time step $t+2$.

Long-term Reward $\hat{\phi}^i(k,T)$: This reflects a heuristic estimate of rewards over steps $t+3$ to $t+T$ if a particular cell choice, k, is made for step $t+2$.

This gives

$$\hat{R}^i(k,t+2) = \hat{\rho}^i(k,t+2) + \lambda\hat{\phi}^i(k,T) \quad (19)$$

where λ is a scaling factor controlling the relative importance given to the long-term reward. Note that λ could, in principle, be UAV-specific and time-varying to allow for the possibility of adaptation. However, we use a fixed value.

Having calculated the estimated reward, $\hat{R}_i(k,t+2)$ for each candidate cell, k, for step $t+2$, the UAV chooses the cell that promises the greatest reward. All UAVs then update their positions synchronously, and each receives the appropriate reward. Note that the action taken at step t was chosen at $t-1$.

The key idea is that, in many cases, the cell that offers the greatest reward at step $t+2$ may not offer the best choices for steps $t+3$ and later. While it would be difficult for a UAV to predict the rewards for these later steps, it could induce some heuristic associations between the current situation in the neighborhood of the targeted cell with future prospects.

3.1 Estimation of the Immediate Reward

In order to make its decision, each UAV, i, must estimate the immediate reward it can expect for each cell, k, that it is considering for step $t+2$. This involves estimating both the coverage and target detection rewards that can be expected at cell k after two time-steps. Next, we describe how this is done.

3.1.1 Estimating the Coverage Reward

Suppose $\nu_k(t+1)$ UAVs occupy cell k at $t+1$ and $\nu_k(t+2)$ (including i) at step $t+2$. Then $z_k(t+1)$ is determined from $z_k(t)$ by iteratively using equation(3) and (4) $\nu_k(t+1)$ times, applying $F_1(\,)$ for all positive sensor measurements and $F_0(\,)$ for all negative ones. Similarly, $z_k(t+2)$ is obtained from $z_k(t+1)$ by iteratively using equation(3) and (4) $\nu_k(t+2)$ times. This can be written as

$$z_k(t+1) = g_1(z_k(t), \nu_k(t+1))$$

and

$$z_k(t+2) = g_2(z_k(t+1), \nu_k(t+2))$$

where $g_1(\ )$ and $g_2(\ )$ are compositions of $F_1(\ )$ and $F_0(\ )$ determined by the string of sensor measurements made at $t+1$ and $t+2$. We assume that UAVs immediately communicate their movement decisions to other UAVs in their vicinity (communicative UAVs) within one time-step, so $\nu_k(t+1)$, which is based on decisions made at time $t-1$, is known at time t, while $\nu_k(t+2)$ is not (since these decisions are made at step t). However, the sensor readings of the UAVs in cell k at $t+1$ and $t+2$ are not known. UAV i, therefore, estimates $z_k(t+1)$ using an "optimistic" heuristic by assuming that each UAV visiting k at $t+1$ makes a reading consistent with $z_k(t)$, i.e., it assumes that $b_t^j = 1$ for all UAVs, j, in cell k if $z_k(t) \geq 0.5$ and 0 if $z_k(t) < 0.5$. This is based on the assumption that $p_s > 0.5$ for all UAVs. In fact, the closer p_s is to 1, the more the heuristic is justified. The estimate, $\hat{z}_k^i(t+1)$, is then used to estimate $z_k(t+2)$. However, this also requires an estimate of $\nu_k(t+2)$, and this is where the UAV uses the adaptive predictor. From this predictor, i obtains a prediction $\hat{\nu}_k^i(t+2)$, and then uses the optimistic heuristic to get $\hat{z}_k^i(t+2)$. Using the estimates, the UAV obtains:

$$\hat{u}_k^i(t+1) = H(\hat{z}_k^i(t+1))$$

$$\hat{u}_k^i(t+2) = H(\hat{z}_k^i(t+2))$$

Finally, it estimates the coverage reward for cell k at time $t+2$ as:

$$\hat{\rho}_c^i(k, t+2) \quad = \quad \frac{\alpha}{\hat{\nu}_k(t+2)}[\hat{u}_k^i(t+1) - \hat{u}_k^i(t+2)] \tag{20}$$

3.1.2 Estimating the Target Detection Reward

The target detection reward is a non-cooperative one, and is simply the expectation that a target will be detected by going to cell k at time $t+2$, provided that $\zeta_k(t+1) = 0$. This quantity can be estimated using approaches from optimal search theory[13]. However, for simplicity, in our model, the target detection reward is estimated simply as:

$$\hat{\rho}_f^i(k, t+2) \quad = \quad z_k(t)[1 - \zeta_k(t)] \tag{21}$$

Essentially, we are using $z_k(t)$ — the current estimate of the target probability in k — as a surrogate for the probability of detecting a target, which is strictly correct only for perfect sensors. More accurate estimation approaches will be reported in future studies. Also, we are using $\zeta_k(t)$ as an estimate for $\zeta_k(t+1)$. An alternative would be to estimate it from $\hat{z}_k^i(t+1)$.

Once the two components of the immediate reward have been estimated, the UAV calculates the immediate reward estimate for cell k as:

$$\hat{\rho}^i(k, t) \quad = \quad \alpha\rho_c^i(k, t+2) + (1-\alpha)\hat{\rho}_f^i(k, t+2) \tag{22}$$

3.2 Estimating Cell Occupancy

The primary estimation problem solved by UAV i in order to decide its move at time $t+2$ is the occupancy, $\nu_k(t+2)$, for all reachable cells, k, where $k \in \{l_i(t+2), f_i(t+2), r_i(t+2)\}$. This is done based on six items of information:

1. **Occupancy information for $t+1$:** $[\nu_{l_i}(t+1), \nu_{f_i}(t+1), \nu_{r_i}(t+1)]$. This is known because UAVs have communicated their moves for $t+1$ by step t

2. **Competition information for $t+2$:** $[c_{l_i}(t+1), c_{f_i}(t+1), c_{r_i}(t+1)]$, with

   $$c_k(t+1) = \frac{1}{\beta}|C_1(x_i^k, y_i^k, t+1)|$$

   where $C_1(x, y, t)$ is the set of UAVs that can reach cell (x, y) in one step after t, $|.|$ denotes cardinality, x_i^k and y_i^k are the coordinates of target cell k, and β is a scaling constant (we use $\beta = 8$). Again, since the positions of neighboring UAVs for $t+1$ are known, $C_i(.,.,t+1)$ can be calculated exactly.

Together, these two sets of values define the *state* for i, $S_i(t) = [\nu_{l_i}(t+1), \nu_{f_i}(t+1), \nu_{r_i}(t+1), c_{l_i}(t+1), c_{f_i}(t+1)c_{r_i}(t+1)]$. We use a neural network consisting of three independent sub-networks to estimate $\nu_{l_i}(t+2)$, $\nu_{f_i}(t+2)$, and $\nu_{r_i}(t+2)$ using $S_i(t)$ as the state input. It should be noted that the state information available to each UAV is an extremely incomplete view of the system's state even in the UAV's neighborhood. More informative state formulations can be envisioned (e.g., certainty values for all neighbors of target cells), but this increases the complexity of the learning problem.

The predicted value of $\nu_k(t+2)$ is used in equation (20) along with the known values of $z_k(t)$ and $\nu_k(t+1)$ to obtain the estimate of the 2-step immediate reward as described above.

3.3 Long-term Reward Estimation

Our approach to estimating the long-term reward term is to use a heuristic based on the average estimated cell occupancy (AECO) ν^i, which is UAV i's current estimate of how many UAVs, on average, occupy any arbitrary cell at a time-step. The UAV uses the heuristic assumption that a cell, h, $T > 2$ steps away will be occupied by ν_i UAVs for each of the next T steps. Since the current target probability value of the cell h, $z_h(t)$, is known, the expected reward for entering that cell T steps later can be estimated just like the immediate reward but iterating $F_0(\,)$ or $F_1(\,)$ ν_i times for each time-step. This estimate is denoted by $\hat{\rho}_i(h, t+T)$.

In considering cell k for step $t+2$, UAV i looks at all cells, h, that can be visited from cell k at step $t+j$, $j \in [3, \ldots, T]$. Let this set be denoted $G_k(j)$. Then, the UAV calculates

$$\hat{\phi}_i(k,T) \quad = \quad \frac{1}{T-2} \sum_{j=3}^{T} \frac{\gamma^{j-3}}{|G_k(j)|} \sum_{h \in G_k(j)} \hat{\rho}^i(h, t+j) \tag{23}$$

where $\gamma \in [0,1]$ is a discount rate and $\hat{\phi}^i(k,T)$ is the discounted average expected rewards for next T steps if cell k is visited at $t+2$. In our current simulations, we use an arbitrary fixed value of $\nu_i = \nu$ for all i. Better estimates are certainly possible. One possibility is for each UAV to maintain a running average of cell occupancies it has encountered in its experience, and to use this as $\nu_i(t)$. Of course, even this is a rather crude estimate. A better estimate could be obtained if the UAV used information about the configuration (or even number) of UAVs in the neighborhood of $G_k(j)$ to make the approach where UAVs start with arbitrary initial values and adapt this using actually observed occupancies and the long-term rewards actually obtained. These methods will be explored in future simulations.

4 Learning Algorithm

As described earlier, the UAVs use tripartite neural networks for predicting $\nu_k(t+2)$, each subnetwork predicting the 2-step occupancy of one of the target cells. This is accomplished with a Q-learning procedure [17], using the true occupancy values, $\nu_k(t+2)$, which become available at time $t+2$. Essentially, the neural networks learns to produce an increasingly accurate estimate of $\nu_k(t+1)$ given $S_i(t)$. The weights of the neural networks are modified using the Levenberg-Marquardt procedure.

As described earlier, we consider three situations:

Centralized Learning (CL): In this case, there is only one tripartite neural network. All UAVs communicate their observations, $\nu_k(t+2)$, to this network, which calculates the errors for all its corresponding predictions and uses these for learning.

Decentralized Learning (DL): In this case, each UAV has its own tripartite network, trained using its own predictions and observations. There is no copying of networks among UAVs.

Opportunistic Cooperative Learning (OCL): In this case, each UAV maintains a running *average* of its prediction quality for occupancy variable $\nu_k(t+2)$. When two UAVs, i and j, find themselves in neighboring cells, they compare their prediction quality values on both variables. UAV i then copies the predictor from UAV j with probability π if the quality of j's prediction is better than its own.

Our previous work [3], using a much simpler search algorithm, has shown that CL provides significantly better performance than DL, presumably because it learns on a more extensive training set. However, the centralized approach scales poorly, requires excessive communication, and is not very robust. We have shown that, on the simple algorithm, the OCL approach has performance approaching that of the CL case, but without the problems of centralization. Th results presented in the next section show that the same is true with the more complex search algorithm used here.

5 Simulation Results

To assess the performance of the approach described above, we simulated a team of four UAVs in an environment with no prior information ($z(0) = 0.5$ for all cells). The UAVs were first trained (using the learning algorithm described above) for T_{train} steps, and then allowed to search an environment without further training. We used two measures of performance:

- Number of targets found up to the current time-step, i,e., the number of cells with $\zeta = 1$.
- The mean residual uncertainty left in the environment:

$$U(t) = \frac{\sum_{(x,y)\in E} u_{x,y}(t)}{\sum_{(x,y)\in E} u_{x,y}(0)} \quad (24)$$

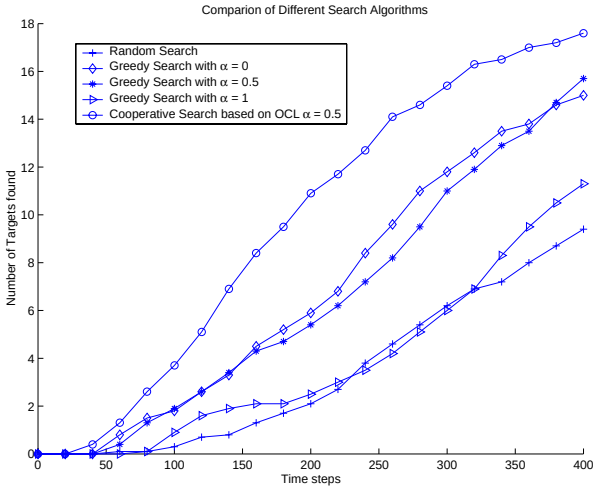

Figure 3: Number of targets found as a function of time: Comparison with greedy and random search algorithms. All data is averaged over 10 runs, and each system was trained for 100 steps. The value of α for the OCL algorithm is 0.5

The performance of the OCL algorithm was compared to that of random search and several greedy algorithms using different values of α. The latter demonstrate the relative utility of the coverage and target detection objectives: The $\alpha = 0$ case corresponds to using only the target objective, the $\alpha = 1$ case to using only the coverage objective, and the $\alpha = 0.5$ case to using both equally. Figure 3 shows the number of targets found by each algorithm as a function of time. Clearly, the cooperative search algorithm does much better. Figure 4 shows the effect of the learning approach on the target detection. In this simulation, it is apprent that, after the initial "easy" uncertainty has been picked up, the CL algorithm does better than the DL algorithm. This reflects the inherent advantages of the centralized approach in terms of learning. However, it is notabl how, after an initial period, the OCL algorithm has performance identical to the CL algorithm even though it is totally decentralized. Figure 5 shows how the mean

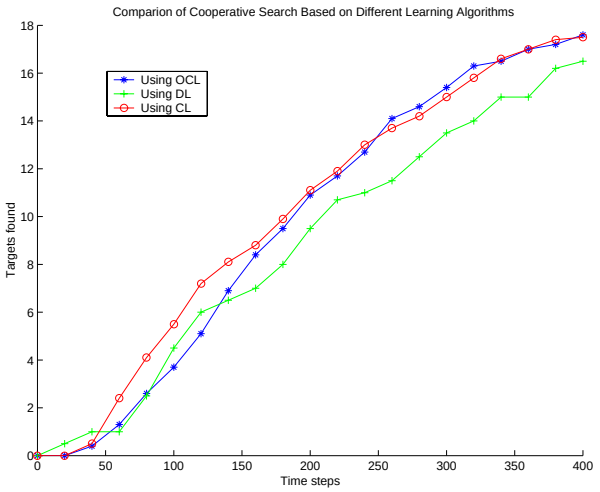

Figure 4: Number of targets found as a function of time: Effect of learning approach. Parameters are as for Figure 3

.

residual uncertainty in the environment declines with time for different search algorithms. Initially, as there is a lot of uncertainty around, all algorithms do well. However, as more and more of the environment is covered, the cooperative search becomes increasingly better at seeking out and covering regions of higher uncertainty.

6 Conclusion

In this paper, we have presented an approach for the cooperative search of an initially uncertain environment using opportunistic cooperative learning. Our results indicate that the approach is a promising one. However, several issues remain to be addressed. These include obtaining better estimators for the reward, and using more intelligent criteria for opportunistic exchange of predictors (e.g., using a variable instead of fixed probability). Also, while we use decentralized learning, the method still uses a centralized TPM. We are currently exploring ways to decentralize the construction and use of the map as well.

Acknowledgement: This research was supported in part by the DARPA/MICA-SHARED program.

References

[1] M. Polycarpou, Y. Yang, and K. Passino. A cooperative search framework for distributed

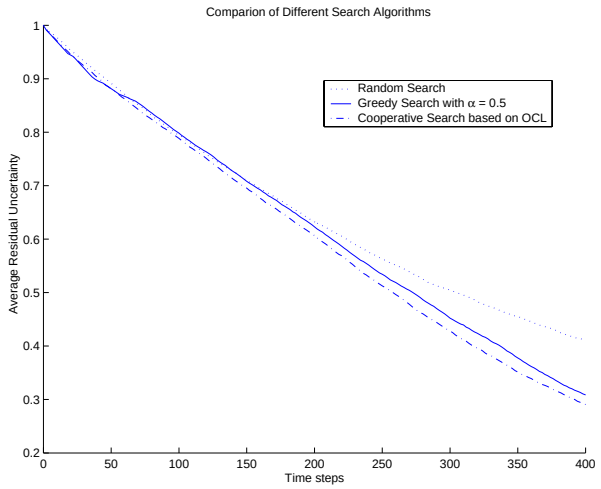

Figure 5: Residual uncertainty as a function of time: Comparison with greedy and random search algorithms. Parameters are as for Figure 3

agents. In *Proceedings of the 2001 IEEE International Symposium on Intelligent Control*, pages 1–6, 2001.

[2] M. Polycarpou, Y. Yang, and K. Passino. Cooperative control of distributed multi-agent systems. *IEEE Control System Magazine*. (submitted).

[3] Y. Yang, M. Polycarpou, and A. Minai. Opportunistically cooperative neural learning in mobile agents. In *Proc. International Joint Conference on Neural Networks, paper no. 2638*, May 2002.

[4] M. Tan. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth International Conference on Machine Learning*, pages 330–337, 1993.

[5] H. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9:61–74, 1988.

[6] K. Nygard, P. Chandler, and M. Pachter. Dynamic network optimization models for air vehicle resource allocation. In *Proc. of the ACC*, pages 1853–1856, June 2001.

[7] P. Chandler, M. Pachter, and S. Rasmussen. Uav cooperative control. In *Proc. of the ACC*, pages 50–55, June 2001.

[8] T. McLain, P. Chandler, S. Rasmussen, and M. Pachter. Cooperative control of uav rendezvous. In *Proc. of the ACC*, pages 2309–2314, June 2001.

[9] R. Smith, M. Self, and P. Cheeseman. Estimating uncertin spatial relationships in robotics. In I.J. Cox and G.T. Wilfong, editors, *Autonomous Robot Vehicles*, pages 167–193. Springer Verlag, 1993.

[10] W. Burgard, D. Fox, M. Moors, R. Simmons, and S. Thrun. Collaborative multi-robot exploration. In *Proc. Intl. Conf. on Robotics and Automation*, May 2000.

[11] I. Rekleitis and E. Milios G. Dudek. Accurate mapping of an unknown world and online landmark positioning. In *Proc. Vision Interface*, pages 455–461, 1998.

[12] P.M. Newman and H.F. Durrant-Whyte. The geometric projection filter — an efficient solution to the slam problem. In G.T. McKee and P.S. Schenker, editors, *Sensor Fusion and Decentralized Control in Robotic Systems: Proc. SPIE Vol. 4571*, pages 23–33, 2001.

[13] L.D. Stone. *Theory of Optimal Search*. Acadamic Press, New York, 1975.

[14] B.O. Koopman. *Search and Screening: General principles with Historical Application*. Pergamon, New York, 1980.

[15] S. Thrun. Learning metric-topological maps for indoor mobile navigation. *Artificial Intelligence*, 99(1):21–71, 1998.

[16] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA.

[17] C. J. C. H. Watkins. *Learning with Delayed Rewards*. PhD thesis, University of Cambridge, 1989.