43rd IEEE Conference on Decision and Control
December 14-17, 2004
Atlantis, Paradise Island, Bahamas

Balancing Search and Target Response in Cooperative UAV Teams

Yan Jin, Yan Liao, Marios M. Polycarpou, Ali A. Minai*
Department of Electrical & Computer Engineering and Computer Science
University of Cincinnati
Cincinnati, OH 45221-0030

***Abstract*— In this paper, we consider a heterogeneous team of UAVs drawn from several distinct classes and engaged in a search and destroy mission over a spatially extended battlefield with targets of several types. Some target locations are suspected *a priori* with a certain probability, while the rest need to be detected gradually through search. The tasks are determined in real-time by the actions of all UAVs and their consequences (e.g., sensor readings), which makes the task dynamics stochastic. The tasks must, therefore, be allocated to UAVs in real-time as they arise. Quick response is more important for known targets, while efficient search is necessary to discover hidden targets. Prediction may help when most targets are known a priori, but could hurt when they are not. In this paper, we study how the benefit of such prediction may depend on the number of targets and UAVs. In particular, we show that there is a trade-off between search and task response in the context of prediction. Based on the results, we propose a hybrid algorithm which balances the search and task response. The performance of proposed algorithms is evaluated through Monte Carlo simulations.**

I. Introduction

Recent advances in intelligent systems and cooperative control have prompted many researchers to study large groups of *unmanned autonomous vehicles* or *UAV*s acting cooperatively to accomplish search and destroy missions in poorly known and hazardous environments ([?], [?], [?], [?], [?], [?]). UAVs engaged in such missions must search for targets and respond to those that are found. In this paper, we look at the delicate balance between these two aspects of the UAVs mission.

We consider a group of UAVs drawn from several distinct classes and engaged in a search and destroy mission over an extended battlefield. The suspected locations of some targets are given a priori, while others must be found through cooperative search by the UAVs. Each suspected target must be confirmed and classified, attacked with appropriate munitions, and have its destruction verified. Since these tasks are created and accomplished through the actions of the UAVs, the task dynamics emerges stochastically over the environment, and requires that tasks be assigned to appropriate UAVs as they arise. This creates a problem similar to the dynamic vehicle routing problem ([?], [?]), but of greater complexity due to the stochastic dynamics and several types of vehicles and tasks.

This work was supported by the AFRL/VA and AFOSR Collaborative Center of Control Science (Grant F33615-01-2-3154). Please address all correspondence to Ali A. Minai (aminai@ececs.uc.edu).

In previous papers, we have reported results from a simplified model where UAVs choose their tasks autonomously and cooperatively using a central cognitive map that provides perfect, instantaneous information on the status of all tasks and UAVs ([?]). In this model, UAVs engage in search as the default behavior, and take on target-specific tasks through a process of *cooperative gradual commitment*, beginning with volunteering and ending in assignment. UAVs assigned to specific tasks proceed directly to the task location instead of searching for new targets. This creates a classic exploration-exploitation trade-off [?], where resources dedicated to search (exploration) compete with those dedicated to target response (exploitation). This trade-off is magnified further when UAVs act on predicted as well as current information, since the predicted tasks create further opportunities for exploitation and take resources away from exploration. In this paper, we present a model significantly more realistic than the one used in our earlier work, and use it to study the issues of predictive assignment and the search-response trade-off with Monte Carlo simulations.

II. Cooperative UAV Team Model

A. Scenario and Model Description

The environment is taken to be a continuous region measuring $L_x\ km$ by $L_y\ km$. For the purposes of sensing and representation, it is divided into $N_x \times N_y$ cells. We use upper case X and Y to denote the integer coordinates of the discretized cellular representation and lower-case x and y for the Cartesian coordinates of the continuous environment.

The environment has M stationary targets, ν_i, $i = 1, \ldots, M$ with locations, (x_i^ν, y_i^ν), and fixed orientations Φ_i^* relative to a globally defined frame of reference. These targets are drawn from N_T different target classes. Of the M targets, M_k are suspected initially, while $M_h = M - M_k$ need to be discovered gradually during search. There are n UAVs, u_i, $i = 1, \ldots, n$, operating in the environment, with the goal of discovering and destroying all targets. A canonical *task set*, $\mathcal{T}$, defines the tasks that the UAVs can undertake at a target location: $\mathcal{T} = \{Search, Confirm, Attack, BDA\}$.

Each UAV, u_i, is characterized by two *expertise vectors*: (1) $\xi_i^S = \{\xi_{ij}^S,\ j = 1, \ldots, N_T\}$, where ξ_{ij}^S indicates the UAV's expertise for sensing and identifying a target of type j; (2) $\xi_i^A = \{\xi_{ij}^A,\ j = 1, \ldots, N_T\}$, where ξ_{ij}^A indicates the UAV's capability for attacking a target of type j. The matrices $\Xi^S = \{\xi_i^S\}$ and $\Xi^A = \{\xi_i^A\}$ are termed the *sensing expertise matrix* and the *attack expertise matrix*, respectively.

UAVs move autonomously through the environment in continuous time, scanning, communicating with other UAVs, making decisions, and performing tasks. At time t, every cell, (X,Y), in the environment has an associated *task status*, $T(X,Y,t) = \{T_j(X,Y,t), j = 0,1,\ldots,N_T\}$, indicating what needs to be done for each possible target type j. Here, $j = 0$ corresponds to the "no target" case. Each T_j can take values 1 (*search*), 2 (*confirm*), 3 (*attack*), and 4 (*BDA*). The task status of all cells, $T(t) = \{T_j(X,Y,t)\}$, represents the *task state* of the environment from the UAV team's viewpoint. The dynamics of the task state is determined by the *target occupancy probability (TOP)*, $P_j(X,Y,t)$, defined as the estimated probability that the cell contains a live target of type j, $j = 1,\ldots N_T$, and $P_0(X,Y,t)$ represents the estimated probability that there is no live target there. It is assumed that there is at most one live target located at a cell and no target crosses the boundary between two or more cells. The TOP of all cells, $P(t) = \{P_j(X,Y,t)\}$, is called the *TOP map*.

The $confirm$, $attack$ and BDA tasks are called *assignable tasks*, i.e., tasks for which the UAVs are assigned explicitly. Such UAVs move purposively to the locations of their assigned tasks and perform them. $search$ is termed an *automatic task*, i.e., any UAV passing through a cell with *search* task status automatically performs search but UAVs do not actively bid for these tasks. The $search$ task does have an effect on UAV movements as described below.

All cells with known assignable tasks at time t form the set, $L(t)$, of *current target locations* (CTLs). The task, τ_i, at each CTL, (X_i,Y_i), has an *assignment status*, A_i, which can take on the values from the set $\mathcal{A} = \{available, associated, assigned, active, complete\}$. The assignment status indicates whether the task is open for bidding (available), has been provisionally assigned to a UAV (associated), has been firmly assigned to a UAV (assigned), is currently being performed by a UAV at the location (active), or has been finished (complete).

B. UAV State

The *state*, $S_i(t)$, of a UAV, u_i, at time t has two parts:

- A *physical state*, including information on its position, $L_i(t)$, speed, $v_i(t)$, heading angle, $\psi_i(t)$, sensor resources, $\zeta^i(t)$, and munition resources, $\mu^i(t)$.
- A *functional state*, indicating the identity and location of the specific task (if any) to which the UAV is committed or has bid for, the corresponding *commitment status* (see below), and the UAV's expected cost for performing this task. The commitment status, $K_i(t)$, of UAV u_i takes values from the set $\mathcal{S} = \{open,\ competing,\ committed\}$, indicating whether the UAV has no commitment (open), has bid on a task or been associated with one (competing), or is assigned to a task and, possibly, is performing it (committed). The functional state of an open UAV has NULL values in its other fields. The $search$ task requires no commitment, and corresponds to an *open* functional state.

C. UAV Kinematic Model

UAVs move on continuous trajectories with constant speed and constraints on turning. For the i^{th} UAV, u_i, the kinematic model is given by:

$$\begin{aligned}\dot{x}_i &= v_i \cos\psi_i \\ \dot{y}_i &= v_i \sin\psi_i \\ \dot{\psi}_i &\leq \eta_1 \\ \dot{v}_i &= 0\end{aligned}$$

where (x_i,y_i) is the position of the UAV, v_i is its speed and ψ_i its heading angle. The third equation specifies the constraint on the turning rate, which cannot exceed η_1. This model has been used widely by other UAV researchers ([**?**], [**?**], [**?**]).

D. Sensor Model

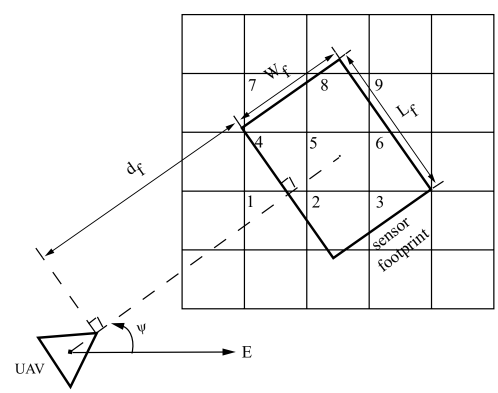

Fig. 1. Sensor model.

Each UAV is assumed to have a fixed rectangular sensor field (footprint) of size $L_f\ km \times W_f\ km$, and $d_f\ km$ ahead of the UAV in the heading direction (Fig. 1). The sensor footprint covers several cells at a time, but only cells whose centers are covered by the footprint are taken as read. In the case of Fig. 1, cells $2,4,5,6,8$ are sensed. All sensors are assumed to have the same sensor field, though this is not essential to the model.

As it traverses the environment performing *search*, a UAV senses at a fixed *sensing rate*, R_s. The sensing rate is chosen relative to UAV speed, v, such that no cell in its path is missed by a UAV travelling in a straight line. If a cell is scanned several times in successive readings, only the first of these is considered. Thus, once a cell is read, it must fall out of the UAV's sensor field before it can be read again. This is to eliminate several highly correlated readings during the same pass over a cell.

Sensor readings taken for *confirm* and *BDA* tasks are handled slightly differently. A UAV proceeding to a cell (X^*, Y^*) for a *confirm* or *BDA* task continues taking sensor readings as for *search*. However, once the target cell enters its sensor field, it ceases these routine readings and waits until the field is centered on the target cell at the desired angle of approach and then takes a reading. The reading still includes all the cells in the field, but has the target cell in the best position. The UAV then refrains from taking further readings until the target cell is out of its sensor field. It then resumes taking regular periodic readings.

E. TOP Dynamics

The TOP map is updated in an event driven fashion by UAV *observations* and *actions*. When UAV u_i takes a sensor reading at time t, observations $b_i(X, Y, t) \in \{0,\ 1,\ ...,\ N_T\}$ are produced for all cells (X, Y) in its sensor field at the time. These are stochastic quantities, with $b_i(X, Y, t) = j$ indicating that UAV u_i detected a target of type j in cell (X, Y) at time t (recall that $j = 0$ corresponds to detecting no target). When a UAV u_i located within cell (X, Y) fires a munition at time t, it is denoted as an action, $a_i(X, Y, t)$. The observations and actions that occur in cell (X, Y) at time t are denoted, respectively, by $b(X, Y, t)$ and $a(X, Y, t)$. Together, they determine the updates of the TOP value at (X, Y) through a possibly stochastic *TOP update function*, F:

$$P(X, Y, t) = F(P(X, Y, t^-), T(X, Y, t^-), a(X, Y, t), b(X, Y, t)) \quad (1)$$

where t^- indicates the time immediately preceding the action or observation. If multiple UAVs take observations or actions in the same cell simultaneously, the updates are applied sequentially.

The TOP update function F is defined as follows for the cases of observation and action:

Observation-Triggered TOP Update Function F_o: If UAV u_i makes a sensor reading $b_i(X, Y, t)$ using sensor resources $\zeta^i(t)$, the TOP for each target type is updated based on the Bayesian formulation ([**?**]):

$$P_j(X, Y, t) = \frac{\lambda^i_{j,b_i(X,Y,t)}(\theta_S(t), \zeta^i(t)) P_j(X, Y, t^-)}{\sum_{l=0}^{N_T} \lambda^i_{l,b_i(X,Y,t)}(\theta_S(t), \zeta^i(t)) P_l(X, Y, t^-)} \quad (2)$$

where θ_S is the *relative angle of observation (RAO)*, given by the angle between the UAV's heading and the target's estimated orientation, and $\lambda^i_{j,k}$ is a function characterizing the accuracy of the sensors used by u_i, defined as:

$$\lambda^i_{j,k}(\theta_S, \zeta^i) = Prob(b_i = k | E_j; \theta_S, \zeta^i).$$

Here, E_j is the event that a target of type j is actually located in the cell being scanned. So $\lambda^i_{j,k}(\theta_S, \zeta^i)$ quantifies the probability of observing a type j target as a type k target from an RAO of θ_S using sensor resources ζ^i. It is assumed that, given a target type and a sensor, there are some optimal angles of observation. A high-quality sensor would have $\lambda^i_{k,k}$ close to 1 for all k when the observation is made from an optimal angle for that target type, but not necessarily from another angle. This models the real situation when the accuracy of a given sensor depends on the type of target and angle of observation. Note that the TOP for all target types in the cell are updated after each observation.

Action-Triggered TOP Update Function, F_a: If UAV u_i with munition resources $\mu^i(t)$ executes an attack in cell (X, Y), the TOP for the cell is updated as:

$$P_j(X, Y, t) = (1 - \beta^i_j(\theta_A(t), \mu^i(t))) P_j(X, Y, t^-), \quad for\ j = 1, 2, \ldots, N_T \quad (3)$$

$$P_0(X, Y, t) = 1 - \sum_{l=1}^{N_T} P_l(X, Y, t) \quad (4)$$

where $0 \le \beta^i_j \le 1$ is the probability that the target of type j is destroyed by UAV u_i (i.e., with the munitions available to it), $\theta_A(t)$ is the *relative angle of attack (RAA)*. As with observation-triggered updates, the TOP for all target types is updated after an action.

F. Task Dynamics

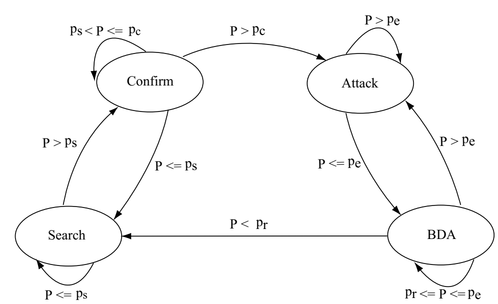

Fig. 2. Automaton formulation of the task dynamics.

Changes in the TOP map determine the dynamics of the cell's task state. This is modelled as a deterministic automaton, H, whose transitions depend on threshold crossings in $P(X, Y, t)$ (Fig. 2):

$$T(X, Y, t) = H(T(X, Y, t^-), P(X, Y, t); \bar{\rho}) \quad (5)$$

where the parameter vector $\bar{\rho}$ represents the set of threshold values used for transitions. The dynamics is made stochastic by the stochasticity of $a(X, Y, t)$ and $b(X, Y, t)$. Fig. 2 shows the transitions between states using an automaton formulation. Details can be found in our previous work ([**?**]).

G. Uncertainty Dynamics

In order to direct the search for targets efficiently, it is important to quantify how much is known about the existence of targets in each cell. We do this by defining an *uncertainty* variable, $\chi(X,Y,t)$. Each observation by a UAV in (X,Y) updates the uncertainty as follows:

$$\chi(X,Y,t) = \omega_\chi[-P_0(X,Y,t)\log P_0(X,Y,t) - (1-P_0(X,Y,t))\log(1-P_0(X,Y,t))] + \frac{(1-\omega_\chi)}{\log N_T}[\sum_{l=1}^{N_T}(-P_l(X,Y,t)\log P_l(X,Y,t))] \quad (6)$$

where, ω_χ is a parameter between 0 and 1. The first term of the equation (6) is the entropy of target existence, while the second term is the entropy of target type. This combined entropy-like formulation provides a measure that represents how certain the UAV team is that a target exists in (X,Y) and about its type.

H. Target Orientation Dynamics

There is a nominal target orientation, $\Phi^*(X,Y)$, associated with the target (if any) in cell (X,Y). However, since the UAVs do not know where and of what type the targets are, each cell, (X,Y), is initialized with a default orientation, $\Phi_j(X,Y,0)$ for each target type, j. $\Phi(X,Y,t)$, which is the collection of all orientation estimates for the environment, comprises the subjective *orientation map* for the UAV team. When UAV u_i makes an observation in (X,Y) using sensor resources $\zeta^i(t)$ and the observation returns $b_i(X,Y,t) = j$, it estimates an orientation, $\phi_t \in [0,2\pi)$ for the type j target. The orientation estimate $\Phi_j(X,Y,t)$ is then updated as:

$$\Phi_j(X,Y,t) = (1-\gamma_{i,j})\Phi_j(X,Y,t^-) + \gamma_{i,j}\phi_t \quad (7)$$

where, $\gamma_{i,j} \in [0,1]$ is the capability of the sensor resource, $\zeta^i(t)$, for estimating the orientation of type j targets (e.g., dependent on nominal variance of the estimate). Note that the target orientation is updated only for the target type detected at time t.

III. Assignment Algorithm

A. Basic Assignment Algorithm

The UAVs' mission is to $search$ all cells for all possible target types, and to perform $confirm$, $attack$ and BDA tasks on each target known or discovered through $search$. For each task, the team must try to use the UAV best suited to it.

All UAVs have instantaneous and noise-free access to a centralized *information base (IB)*, which comprises the TOP map, orientation map, uncertainty map, task status map, assignment status map and UAV state vector. Each UAV reads and updates the Information Base continuously.

Initially, if a cell, (X,Y), is suspected to have a target of type j^* (at most one), $P_j(X,Y,0)$ is assigned as follows:

$$P_j(X,Y,0) = \begin{cases} P_{j^*}(X,Y,0) \geq p_s, & \text{if } j = j^* \\ \frac{1-P_{j^*}(X,Y,0)}{N_T}, & \text{otherwise} \end{cases},$$

and the corresponding task status is set as:

$$T_j(X,Y,0) = \begin{cases} confirm, & \text{if } j = j^* \\ search, & \text{otherwise} \end{cases}.$$

For those cells which have no suspected target of any type, $P_j(X,Y,0) = \frac{1}{1+N_T}$, and $T_j(X,Y,0) = search$, for $j = 0,1,2,\ldots,N_T$.

The UAVs' initial positions are also given. All UAVs initially have the *open* status. The current set of assignable tasks is $T_s = \{\tau_k\}$, and m_k denotes the *identity* of task τ_k, i.e., whether it is to confirm a target of type j ($m_k = (j,2)$), to attack a target of type j ($m_k = (j,3)$), or to do BDA on a target of type j ($m_k = (j,4)$). The initial assignment is done as follows:

Each UAV u_i calculates a *cost value*, h_{ik}, with respect to all *available* or *associated* assignable tasks, τ_k:

$$h_{ik} = \omega_c d_{ik} + (1-\omega_c)\exp(-\xi_{im_k}) \quad (8)$$

where, ω_c is a parameter valued in [0,1], d_{ik} is the normalized distance between UAV u_i and the location of task τ_k, and ξ_{im_k} is the expertise of UAV u_i for task m_k. UAV u_i is eligible for task m_k if the expertise $\xi_{im_k} \geq \xi_{min}$, where ξ_{min} is a non-negative parameter.

Each UAV reports its cost for all tasks on the CTL for which it is eligible to the central controller, which then uses a semi-greedy bipartite matching algorithm to match UAVs with tasks. UAVs that are within distance D_a of their matched task are $assigned$ the task and are given the $committed$ status, while UAVs that are further away are $associated$ with their matched tasks and are given the $competing$ status. We allow only one UAV to be $assigned$ to a task but up to n_a UAVs can be $associated$ with a task. Similarly, each UAV can only be $committed$ to a single task, but we allow it to be $competing$ for up to m_a tasks. When a UAV has no task to choose, it has $open$ status and follows a *path of maximum local uncertainty*, i.e., one that takes it through cells with the highest uncertainty in its local neighborhood — within turning constraints. The purpose is to maximize the benefit from *search* in a greedy way, and the path followed is termed a *search path*. UAVs assigned to a task determine the best RAO (or RAA) with respect to their sensors (or munitions) and plan a path to approach the target from that angle.

After the initial assignment, each UAV with an $assigned$ task moves towards that task, UAVs with no $assigned$ task move towards their lowest-cost $associated$ task, while the rest follow search paths. All UAVs take sensor readings as they move and update the TOP. When a UAV reaches its $assigned$ task, it performs the task and updates the TOP there. A new task (possibly the same as the previous one) is then cued at the CTL according to the transition function, and the UAV's status reverts to $open$. Locations can become CTLs if $search$ raises their TOP above p_s, corresponding to the discovery of a new target. Each new assignable task is cued with an $available$ status.

At all times, all $open$ and $competing$ UAVs monitor the CTL, and report their costs for all $available$ and $associated$ tasks. When a $competing$ UAV reaches a point within distance D_a of its $associated$ task, that task is $assigned$ to it and its status is switched to $committed$.

All other UAVs $competing$ for this task are dis-$associated$ from it. The controller continually monitors all the costs reported, and allows as many as n_a $competing$ or $open$ UAVs to be $associated$ with it while ensuring that no UAV is $associated$ with more than m_a different tasks. Sometimes, a task may be done opportunistically by a UAV that happens to pass by, in which case all UAVs $assigned$ to or $associated$ with it are released. The process continues until the region is completely searched and all targets are neutralized, or some time limit is reached.

B. Predictive Assignment Algorithm

The assignment algorithm described above uses only the list of currently active tasks for assignment. However, given the task transition thresholds and current assignments, it is possible to predict the next set of tasks and to estimate their probabilities. The process can be iterated to predict tasks further in the future, albeit with decreasing certainty. Including the predicted tasks in the assignment procedure can potentially allow the UAVs to plan their commitments early, and provide assignments for UAVs that cannot do any currently available task. An assignment algorithm using predicted tasks is described below:

A set of *predicted task locations (PTLs)* is formed containing all the CTLs with $associated$ or $assigned$ tasks. For a PTL (X, Y), let u_{i^*} be the UAV that is $assigned$ to or $associated$ with this task (if the task has more than one $associated$ UAV, u_{i^*} is the nearest one among them). The UAV's current sensor and munitions resources are denoted as $\zeta_{i^*}(t)$ and $\mu_{i^*}(t)$, respectively, and θ_{i^*} is its planned approach angle for the current task. Knowing these parameters, one can estimate the time, t^*, at which the UAV will accomplish this task. This is termed the *estimated completion time (ECT)* for the task. Also using the update equations (2), (3) and (4), one can estimate the TOP $\overline{P}(x, y, t^*)$ after the task is completed. The task transition function (Fig. 2) can then be used to get a list of potential successor tasks, $\overline{\tau}_j$, and their probabilities, π_j, for $j = 1, 2, \ldots, N_T$.

All $open$ UAVs then consider bidding for the predicted tasks according to their distance from the task and capabilities. The commitment procedure for predicted tasks is the same as for current ones. Consider a UAV, u_i, with location, (x_i, y_i), sensing (or attacking) expertise profile, $\xi_i(t)$. For a PTL, (X, Y), the UAV, u_i, calculates the minimum time, $t_{i,(x,y)}$, needed to reach it. The time cushion, $\delta_{i,(x,y)}$, is defined by $\delta_{i,(x,y)} = (t^* - t) - t_{i,(x,y)}$. If $\delta_{i,(x,y)} \geq 0$, the UAV, u_i, would reach the PTL, (x, y), before the successor task becomes available, and it will loiter there until the successor task becomes available. If $\delta_{i,(x,y)} < 0$, the task is likely to be available by the time the UAV gets to the PTL. We define a function $Q(z)$ as:

$$Q(z) = \begin{cases} 0, & \text{if } z \geq 0 \\ |z|, & \text{otherwise} \end{cases} .$$

The UAV's cost value with respect to the predicted tasks at PTL (X, Y), is calculated as:

$$h^P_{ik} = \omega_c Q(\overline{\delta}_{i,(x,y)}) + (1 - \omega_c) \exp -(\pi(x, y, t^*) \cdot \xi_i(t))$$

where $\overline{\delta}_{i,(x,y)}$ is the normalized time cushion value.

The assignment process is the same as that in the basic assignment algorithm. Once a UAV is associated with the predicted tasks at a PTL, the successors of those tasks can be predicted.

IV. Simulation Results and Discussion

To evaluate the performance of the basic and predictive assignment algorithms, we conducted Monte Carlo simulations using an event-driven simulator. In the simulations, we consider two types of targets, type 1 and type 2, which are characterized by differences in their optimal angles of detection and attack. The UAVs are drawn from two classes: *target recognition (TR)* UAVs and *attack (A)* UAVs. All UAVs are assumed to have sensors needed for search, but with different sensing capabilities.

The goal for the UAV team is to completely search the region and neutralize all targets as rapidly as possible. In this paper, however, we focus mainly on how the use of prediction affects the time until all targets are *actually* destroyed. Thus, we use this time, termed the *target neutralization time (TNT)* as the measure of performance.

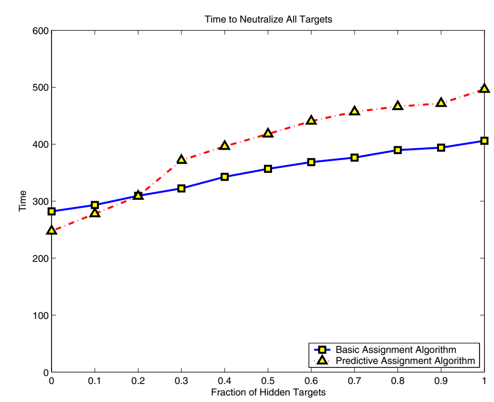

Fig. 3. TNT vs Hidden Target Fraction. 5 TR and 5 A UAVs.

The first set of simulation results (Fig. 3) compare the basic and predictive algorithms for a team of 5 TR and 5 A UAVs in an environment with 10 targets. The number of targets whose positions are suspected *a priori* is varied from 10 (no hidden targets) to 0 (all targets hidden). The graph clearly shows the crossover predicted by the hypothesized search-response trade-off. When the UAVs know almost all the target locations from the start, complete neutralization is achieved faster by using prediction — presumably because UAVs get in position for future tasks early rather than "wasting" time on $search$. However, when the fraction of targets with known locations decreases, $search$ becomes more crucial to the mission, and the predictive algorithm falls behind the non-predictive one.

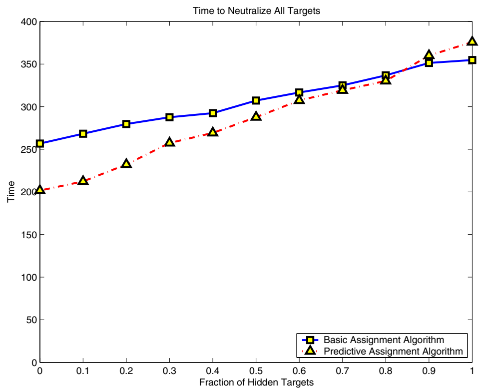

Fig. 4. TNT vs Hidden Target Fraction. 8 TR and 8 A UAVs.

In the second simulation (Fig. 4), the UAV team is comprised of 8 TR and 8 A UAVs in an environment with 10 targets. Thus, the number of UAVs is higher relative to the number of targets than in the situation shown in Fig. 3. This time, prediction provides a significant advantage until the number of hidden targets reaches 7 out of 10. This is consistent with our hypothesis that prediction is most useful when the UAV team has sufficient resources to exploit it. Since this time there are more UAVs in the team than before, only part of the team is needed to handle current tasks, leaving the rest to get in position for predicted tasks, thus reducing the overall TNT. However, this does not neutralize the essential search-response trade-off, and when almost all targets are hidden, the loss of $search$ resources due to predictive assignment does become a liability.

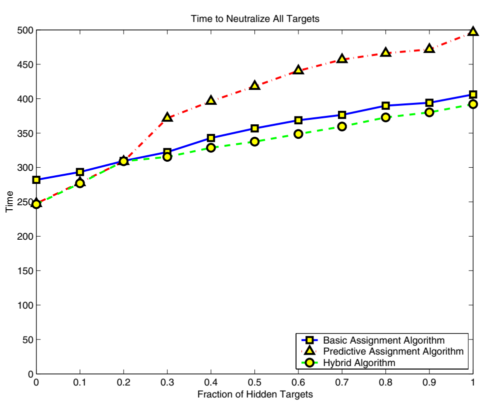

Fig. 5. TNT vs Hidden Target Fraction. 5 TR and 5 A UAVs.

Finally, Fig. 5 shows results for a hybrid algorithm where UAVs start out in the non-predictive mode but switch to the predictive algorithm once a sufficient number of targets have been found. The switching point is obtained using the data in Figure 3. The results show that this algorithm captures the best of both the predictive and the non-predictive approaches. However, it requires that the UAVs know the total number of targets *a priori* — though the positions of some are still unknown and must be found through search. We are investigating algorithms where other, more plausibly available information can be used to affect the switch. In particular, we are investigating methods by which the UAV team can estimate its degree of ignorance about the environment relative to the available resources, and use this as the basis of switching between predictive and non-predictive assignment.

V. Conclusion

We have shown that prediction can help improve the performance of cooperative UAV teams engaged in search-and-destroy missions. Our results show that the utility of prediction depends significantly on the UAV team's size relative to the number of targets, and the UAVs' knowledge of the target locations. Our simulations also show that there is a well-defined crossover point as this knowledge changes. The dependence of this crossover on various system parameters must be studied in greater detail in order to develop practical adaptive algorithms.

References

[1] R. Beard and T. McLain, "Multiple UAV cooperative search under collision avoidance and limited range communication constraints," in *Proc. of the 42nd IEEE Conference on Decision and Control*, Maui, HI, Dec 2003, pp. 25–30.

[2] A. Gil, K. Passino, and A. Sparks, "Cooperative scheduling of tasks for networked uninhabited autonomous vehicles," in *Proc. of the 42nd IEEE Conference on Decision and Control*, Maui, HI, Dec 2003), pp. 522–527.

[3] M. Flint, E. Fernandez-Gaucherand, and M. Polycarpou, "Cooperative control for UAV's searching risky environments for targets," in *Proc. of the 42nd IEEE Conference on Decision and Control*, Maui, HI, Dec 2003, pp. 3567–3572.

[4] S.-M. Li, J. Boskovic, S. Seereeram, R. Prasanth, R. Amin, R. Mehra, and R. a. M. T. Beard, "Autonomous hierarchical control of multiple unmanned combat air vehicles (UCAVs)," in *Proc. of American Control Conference*, 2002, pp. 274–279.

[5] P. Chandler, M. Pachter, and S. Rasmussen, "UAV cooperative control," in *Proc. of American Control Conference*, 2001, pp. 50–55.

[6] Y. Yang, A. Minai, and M. Polycarpou, "Decentralized opportunistic learing in UAV's performing cooperative search," in *Proc. of AIAA Guidance, Navigation, and Control Conference*, 2002.

[7] T. Schouwenaars, B. De Moor, E. Feron, and J. How, "Mixed integer programming for multi-vehicle path planning," in *Proc. of European Control Conference*, 2001, pp. 2603–2608.

[8] K. Tan, L. Lee, Q. Zhu, and K. Ou, "Heuristic methods for vehicle routing problem with time windows," *Intelligent in Engineering*, pp. 281–295, 2002.

[9] Y. Jin, A. Minai, and M. Polycarpou, "Cooperative real-time search and task allocation in UAVs teams," in *Proc. of the 42nd IEEE Conference on Decision and Control*, Maui, HI, Dec 2003, pp. 7–12.

[10] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*, ser. Adaptive Computation and Machine Learning. MIT Press, March 1998.

[11] J. Finke, K. Passino, and A. Sparks, "Cooperative control via task load balancing for networked uninhabited autonomous vehicles," in *Proc. of the 42nd IEEE Conference on Decision and Control*, Maui, HI, Dec 2003, pp. 31–36.

[12] R. Beard, T. McLain, M. Goodrich, and E. Anderson, "Coordinated target assignment and intercept for unmanned air vehicles," *IEEE Trans. Robot. Automat.*, vol. 18, pp. 911 –922, 2002.

[13] P. Chandler, S. Rasmussen, and M. Pachter, "UAV cooperative path planning," in *Proc. of AIAA Guidance, Navigation, and Control Conference*, 2000, pp. 1255–1265.

[14] H. Moravec, "Sensor fusion in certainty grids for mobile robots," *Artificial Intelligence Magazine*, vol. 9, pp. 61–74, 1988.