Balancing Search and Target Response in Cooperative Unmanned Aerial Vehicle (UAV) Teams

Yan Jin, *Student Member, IEEE*, Yan Liao, Ali A. Minai, *Member, IEEE*, and Marios M. Polycarpou, *Fellow, IEEE*

***Abstract*—This paper considers a heterogeneous team of cooperating unmanned aerial vehicles (UAVs) drawn from several distinct classes and engaged in a search and action mission over a spatially extended battlefield with targets of several types. During the mission, the UAVs seek to confirm and verifiably destroy suspected targets and discover, confirm, and verifiably destroy unknown targets. The locations of some (or all) targets are unknown *a priori*, requiring them to be located using cooperative search. In addition, the tasks to be performed at each target location by the team of cooperative UAVs need to be coordinated. The tasks must, therefore, be allocated to UAVs in real time as they arise, while ensuring that appropriate vehicles are assigned to each task. Each class of UAVs has its own sensing and attack capabilities, so the need for appropriate assignment is paramount. In this paper, an extensive dynamic model that captures the stochastic nature of the cooperative search and task assignment problems is developed, and algorithms for achieving a high level of performance are designed. The paper focuses on investigating the value of predictive task assignment as a function of the number of unknown targets and number of UAVs. In particular, it is shown that there is a tradeoff between search and task response in the context of prediction. Based on the results, a hybrid algorithm for switching the use of prediction is proposed, which balances the search and task response. The performance of the proposed algorithms is evaluated through Monte Carlo simulations.**

***Index Terms*—Cooperative search, path planning, task allocation, unmanned aerial vehicle.**

I. Introduction

OVER the last decade, unmanned vehicles such as airborne drones [1] and minesweeper robots have become an increasingly feasible component of the battlefield environment and may soon be common in civilian applications such as disaster relief, environmental monitoring, and planetary exploration. The European project Civil UAV APplications and Economic effectivity of potential CONfiguration solutions (CAPECON) has been launched in 2001 to identify civil UAV applications and define civil UAV configurations [2]–[4].

Manuscript received September 26, 2004; revised May 5, 2005. This work was supported by the Air Force Research Laboratory Air Vehicle Directorate (AFRL/VA) and Air Force Office of Scientific Research (AFOSR) Collaborative Center of Control Science under Grant F33615-01-2-3154. This paper was recommended by Associate Editor D. Y. Lee.

Y. Jin, Y. Liao, and A. A. Minai are with the Department of Electrical & Computer Engineering and Computer Science, University of Cincinnati, Cincinnati, OH 45221-0030 USA (e-mail: aminai@ececs.uc.edu).

M. M. Polycarpou was with the Department of Electrical & Computer Engineering and Computer Science, University of Cincinnati, Cincinnati, OH 45221-0030 USA. He is now with the Department of Electrical & Computer Engineering, University of Cyprus, Nicosia 1678, Cyprus.

Digital Object Identifier 10.1109/TSMCB.2005.861881

However, the unmanned vehicles in current use, such as the Predator, are not autonomous, requiring remote guidance by a team of human operators. Not only is this expensive and risky, it also places a fundamental limit on the scalability and range of such systems. Recent advances in intelligent systems and cooperative control have prompted many researchers to study large groups of UAVs acting cooperatively to accomplish missions in uncertain and hazardous environments [5]–[17]. Most research in this field considers UAV teams engaged in search-and-response missions, i.e., the UAVs must search for targets and respond to those that are found. Efficient search and appropriate task assignment are two crucial components of this problem.

A. Related Work on the Cooperative Search Problem

A significant amount of research on search path planning can be found in the robotics literature in the field of robot motion planning [18] and, in particular, within the subfields of terrain acquisition [19]–[21] and coverage path planning [22]–[24]. Typically, robot motion planning considers a poorly known region populated with unknown but stationary obstacles [25]–[27]. The robot is equipped with a sensing capability that can identify the regions of opportunity and regions with obstructions. The objective is to design algorithms that enable the robot to generate a shortest path through this terrain such that the sensors can scan every point in it. Applications include floor painting (i.e., robotic systems that paint all reachable areas on the floor in a building using the minimum amount of paint), mine clearing [28], lawn mowing [29], seed spreading, room vacuuming [30], field plowing, terrain acquisition [25]–[27], and, of interest to us, complete search for a target [31]–[33]. An excellent survey of the major results in search theory is available in [34]. The search algorithms utilized in robot motion planning provide a good source of intuition for search operations by UAVs. However, most of these algorithms have not been integrated with task allocation in the case of stochastic task transitions.

In recent years, there has been a great deal of work on search schemes for UAVs. A map-based strategy for cooperative search by a team of mobile UAVs is presented in [35]. The basic idea is that each UAV maintains a cognitive map of the search region, assigning an uncertainty measure to each cell within this region. Based on the information in its cognitive map, each UAV calculates and updates its search path to minimize the total uncertainty in the region. The opportunistic neural learning method [36]–[38] and the dynamic programming technique in [39] and [40] are examples of this map-based approach. A

search-theoretic approach based on "rate of return" maps is proposed in [41]. The idea of surrogate optimization, which is a non-gradient-based nonlinear programming method, has also been used to solve this problem [42]. In this approach, each UAV adaptively identifies a terrain map and uses it to obtain decentralized controllers that, in turn, produce its trajectories in a coordinated manner. Genetic algorithms have been applied to guide the search behavior of UAVs, e.g., the SAMUEL evolutionary learning system [43] developed by the Naval Research Laboratory has been used to create rules to guide UAV [44]. In [45] and [46], the cooperative search problem is posed as a pursuit–evasion game.

B. Related Work on the Task Assignment Problem

The task assignment problem deals with the design of algorithms for coordinated assignment of tasks to different UAVs. The task assignment problem for UAVs is similar to the general vehicle routing problem (VRP), which is a well-known nonpolynomial-hard (NP-hard) combinatorial optimization problem in computer science [47]. Many heuristic algorithms have been proposed for this problem. Genetic algorithms with different crossover policies have been proposed [48]–[50]. A detailed survey of heuristic methods for VRP with time windows has been given in [51]. However, the assignment problem for a UAV team is of greater complexity due to its stochastic dynamics, as well as dimensions and difficulties not present in typical ground-based applications confined to roadmaps.

The UAV task assignment problem without stochastic task transitions can be modeled as a large optimization problem, such as a single mixed-integer linear programming problem [52]. This method is computationally intensive, but it is guaranteed to get the globally optimal solution of the assignment problem and can work as a benchmark against which approximate approaches or heuristic methods can be compared. Tabu search has been employed successfully to find the near optimal cooperative assignment for UAV teams minimizing the total traveling time [53] and maximizing the expected target coverage [54]. These methods include fixed time windows for visiting each task location but do not consider multiple successor tasks at each target location or hidden targets.

A stochastic version of the weapon assignment problem has been addressed using stochastic programming in [55]. In the formulation, a set of weapons can be fired at a set of available targets, but some targets may not be known initially. The objective is to maximize the expected destroyed target value over the whole engagement. A stochastic integer program is solved to balance the value of firing weapons at the detected targets against the value of holding weapons to fire at undetected targets. This formulation assumes that there are no time constraints and that each weapon can only be fired against one target.

The task assignment problem has also been viewed as a cooperative scheduling (resource allocation) problem in [56], where the cooperation performance is investigated in the presence of imperfect communications (e.g., messages with random but limited delays). A stability analysis of this network-based cooperative resource allocation strategy is given in [57].

Approaches to the allocation problem, which emphasize timing constraints, have also been proposed [58]–[61]. In this approach, detailed paths are selected for each UAV so as to guarantee simultaneous arrival at an antiaircraft defense system while minimizing exposure to radar along the path. This is performed through the use of coordination variables: Each UAV calculates its own minimum arrival time as a function of radar exposure and broadcasts it to the team. Each UAV in the team then solves the same optimization problem to determine the arrival time, which minimizes radar exposure and allows all UAVs to arrive simultaneously. Schumacher *et al.* [62] address the task assignment problem with timing constraints in a UAV team with geographically dispersed targets. The problem is solved using mixed-integer linear programming, which is guaranteed to find a solution if the UAVs have sufficient endurance.

Research has also focused on suitable control architectures for the coordination of UAV teams, especially in the context of the Low Cost Autonomous Attack System (LOCAAS) [63]. LOCAAS is a miniature autonomous powered munition capable of broad area search, identification, and destruction of a range of mobile ground targets. A hierarchical control strategy has been proposed in [64], which decomposes this problem into three levels, namely 1) team formation; 2) intrateam task allocation; and 3) individual UAV control. Decision making methods at each of these levels have been discussed in [65]. A limited-horizon minimum-weight spanning tree is used to partition the UAVs. An iterative network flow method is applied to make the intrateam assignment. At each iteration, this method carries out a temporary assignment of all the remaining tasks among the UAVs first and then fixes the assignment with the earliest estimated time. This process is repeated until all the tasks are assigned. The allocation problem has also been solved using dynamic network flow optimization models [16] and market-based allocation methods [66].

C. Overview of the Present Paper

As discussed above, the problem of multiagent coordination in general and of UAV teams in particular has received significant attention. However, the research we report makes four important contributions in this area.

1) The mission model we describe and use goes well beyond the model used in the UAV area in several ways.
   a) The search and response problems are considered together, allowing the tradeoff between them to be explored.
   b) Both known and hidden task locations are included.
   c) Task transitions are modeled stochastically, and occur in response to UAV actions rather than deterministically or autonomously.
   d) UAVs are not assumed to be identical, and are modeled as having different—possibly even time-varying—capabilities for each task.
   e) Erroneous sensor readings on the part of UAVs are modeled explicitly in a parameterized way.
   f) UAV and target orientations are modeled explicitly and are related to outcomes.

Other researchers have included some of these features in their models, but by including all of them, our model captures the complexities of the general problem more completely. In particular, by stochastically linking the state of the unfolding mission to the actions of the UAVs, we are able to evaluate the performance of the UAV team more realistically than models where tasks are known *a priori* and scheduling is the only issue [56].

2) As in most work on multiagent assignment [56], [67], [68], we use a greedy approach to task selection by individual UAVs but mitigate the effects of this simplification by using a gradual commitment scheme and allowing cooperative reassignment.
3) Because we consider both search and task response within a single framework, we are able to evaluate the tradeoff between them in assessing the value of prediction. This is an important question, and it has been suggested that, given the highly uncertain nature of most UAV missions, prediction may not be of much value [56]. We address this issue concretely by considering the actual costs and benefits of prediction in terms of the tasks that UAVs must perform.
4) Based on our investigations, we present a hybrid algorithm that allows UAVs to balance the needs of search and task response in an adaptive way to obtain the full benefit of prediction.

The remainder of this paper is organized as follows. In the next section, the scenario of the research problem is introduced. The UAV team model and target information dynamics are defined in Sections III and IV, respectively. In Section V, the basic and predictive assignment algorithms are proposed. Monte Carlo simulation results are presented and discussed in Section VI. Finally, Section VII concludes the paper with a summary and a brief discussion of extensions and future direction for our work.

II. 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. Thus, each cell is L_x/N_x km by L_y/N_y km. Uppercase X and Y denote the integer coordinates of the discretized cellular representation and lowercase x and y represent 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. It is assumed that the UAVs do not know the total number of targets. The UAV team keeps track of current information about targets using an occupancy grid—an approach widely used in robotics [69].

The set of tasks that UAVs can perform at each cell comprise the canonical task set $\mathcal{T}$. In this paper, we consider the task set

$$\mathcal{T} = \{\text{search}, \text{confirm}, \text{attack}, \text{BDA}\}.$$

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 in that cell. Here, $j = 0$ corresponds to the "no target" case. Each T_j can take values 1 (search), 2 (confirm), 3 (attack), or 4 (BDA). Thus, $T_j = 1$ means "search for target of type j," while $T_j = 3$ means "attack target of type j." 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) $P(X, Y, t) = \{P_j(X, Y, t)\}$ of each cell (X, Y) defined as the estimated probability that the cell contains a live target of type j, $j = 1, 2, \ldots, N_T$. $P_0(X, Y, t)$ represents the estimated probability that there is no live target in the cell (X, Y). In the current model, it is assumed that there is at most one live target located in a cell and that no target crosses the boundary between two or more cells. However, since the UAVs do not know about the type (or even existence) of targets *a priori* and obtain their information using imperfect sensors that can confuse between target types, we typically have $0 < P_j(X, Y, t) < 1$ for all (X, Y) and j. The TOP of all cells $P(t) = \{P_j(X, Y, t)\}$ is called the TOP map of the environment and represents the UAV team's subjective estimate of target occupancy throughout the environment. The TOP map is updated as cells are scanned during the mission.

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 $\Phi(X, Y, t) = \{\Phi_j(X, Y, 0)\}$. These orientation estimates are updated whenever a target is detected by the sensors (see below). The set of orientation estimates over the whole environment $\Phi(t) = \{\Phi(X, Y, t)\}$ constitutes the subjective orientation map for the UAV team.

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. The search task is termed an automatic task, i.e., any UAV passing through a cell with search task status automatically performs search when it scans the cell with its sensors, but the task is not explicitly assigned to a specific UAV. 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

$$\mathcal{A} = \{\text{available}, \text{associated}, \text{assigned}, \text{active}, \text{complete}\}.$$

The assignment status indicates whether the task is open for assignment (available), has elicited interest from 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). A completed task is accompanied by an immediate transition in the task status of the CTL, possibly to the same task if it needs to be repeated.

III. UAV Team Model

A. UAV State

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

1) physical state, including information on its position $(x_i(t), y_i(t))$, speed $v_i(t)$, heading angle $\psi_i(t)$, sensor resources $\zeta^i(t)$, and munition resources $\mu^i(t)$;
2) functional state, indicating the identity and location of the specific task (if any) to which the UAV is assigned or associated with, 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 on values from the set

$$\mathcal{S} = \{\text{open}, \text{competing}, \text{committed}\}$$

indicating whether the UAV has no commitment (open), has volunteered for 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.

B. UAV Kinematic Model

UAVs move on continuous trajectories with constant speed and constraints on turning. For the ith UAV u_i, the kinematic model is given by

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

where (x_i, y_i) is the position of the UAV, v_i is its speed, and ψ_i is its heading angle. The third equation specifies the constraint on the turning rate, which cannot exceed η_1. This model has been used by other UAV researchers as well [58], [59], [70], and [71].

C. UAV Sensor Model

Each UAV is assumed to have a fixed rectangular sensor field (footprint) of size L_f km by W_f km and d_f km ahead of the UAV in the heading direction (as shown in Fig. 1). Here, it is assumed that all the UAVs know the position of the center of each cell using a localization system [e.g., global positioning system (GPS)]. The sensor footprint covers several cells at a time, but only cells whose centers are covered by the footprint are taken as read. For example, in the case of Fig. 1, cells 2, 4, 5, 6, and 8 are sensed. All sensors are assumed to have the same sensor field, though this is not essential to the model.

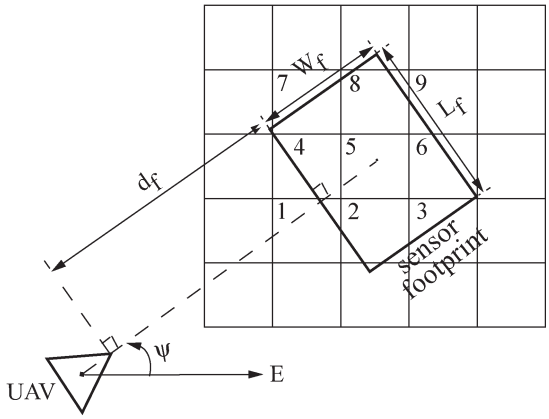

Fig. 1. UAV's sensor model.

As it traverses the environment performing search, a UAV senses at a fixed sensing rate R_s so a new reading of the cells currently covered by the sensor field is taken every $1/R_s$ s. The sensing rate is chosen relative to UAV speed, the size of sensor footprint, and the size of the cell, such that no cell in its path is missed by a UAV traveling 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 eliminates 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.

D. UAV Capabilities

Each UAV u_i is characterized by two capability vectors.

1) Sensing capability vector: $\xi_i^S(t) = \{\xi_{ij}^S(t)\}$, $j = 1, 2, \ldots, N_T$. Each $\xi_{ij}^S(t)$ is determined by the UAV's sensor resources $\zeta^i(t)$ and indicates the UAV's expertise for sensing and identifying a target of type j.
2) Attack capability vector: $\xi_i^A(t) = \{\xi_{ij}^A(t)\}$, $j = 1, 2, \ldots, N_T$. Each $\xi_{ij}^A(t)$ is determined by the UAV's munition resources $\mu^i(t)$ and indicates the UAV's capability to attack a target of type j.

The matrices $\Xi^S = [\xi_{ij}^S]$ and $\Xi^A = [\xi_{ij}^A]$ are termed the sensing capability matrix and the attack capability matrix, respectively. The sensing and attack capability vectors are taken to be time varying in general and can be used to model

in-field changes in UAV capabilities due to sensor damage and munitions use.

1) Sensing Capability: For UAV u_i, $\lambda^i_{j,k}$ is a function characterizing the accuracy of the sensor resources $\zeta^i(t)$, used by u_i, which is defined as

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

where E_j is the event that a target of type j is actually located in the cell being scanned, and $\theta_S(t)$ is the relative angle of observation (RAO), which is defined as the angle between the UAV's heading and the estimated orientation of the type j target in the cell. Therefore, $\lambda^i_{j,k}(\theta_S(t),\zeta^i(t))$ quantifies the probability of observing a type j target as a type k target from an RAO of $\theta_S(t)$ using sensor resources $\zeta^i(t)$. Guided by practical experience in target detection, 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.

In the current model, the sensing capability $\xi^S_{ij}(t)$ is defined as

$$\xi^S_{ij}(t)=\begin{cases}1-\exp\left(-\Gamma^i_j\left(\zeta^i(t)\right)\right), & \text{if } \Gamma^i_j\left(\zeta^i(t)\right)\geq 1\\ 0, & \text{otherwise}\end{cases}$$

where

$$\Gamma^i_j\left(\zeta^i(t)\right)=\max_{\theta\in[0,2\pi)}\frac{\lambda^i_{j,j}\left(\theta,\zeta^i(t)\right)}{\sum\limits_{l=0,\,l\neq j}^{N_T}\lambda^i_{j,l}\left(\theta,\zeta^i(t)\right)}=\max_{\theta\in[0,2\pi)}\frac{\lambda^i_{j,j}\left(\theta,\zeta^i(t)\right)}{1-\lambda^i_{j,j}\left(\theta,\zeta^i(t)\right)}$$

$\lambda^i_{j,j}(\theta,\zeta^i(t))$ quantifies the probability of correct detection, thus, $\Gamma^i_j(\zeta^i(t))$ indicates the UAV's best sensing expertise achievable using the sensor resources $\zeta^i(t)$. If $\Gamma^i_j(\zeta^i(t))<1$, which means that the sensor is likely to make a false reading, the UAV is considered ineligible for sensing type j targets and $\xi^S_{ij}(t)=0$. This is a purely phenomenological model intended to map the $[0,\infty)$ range of the correct detection odds to the [0, 1] range for capability. Other specific models can be substituted without changing the overall framework.

2) Attack Capability: For UAV u_i, β^i_j is a function characterizing the attacking capability of the munition resources $\mu^i(t)$, used by u_i, which is defined as

$$\beta^i_j\left(\theta_A(t),\mu^i(t)\right)=\text{Prob}\left(\text{attack destroys target}|E_j;\theta_A(t),\mu^i(t)\right)$$

where E_j is the event that a target of type j is actually located in the cell being attacked, and $\theta_A(t)$ is the relative angle of attack (RAA), which is defined as the angle between the UAV's heading and the estimated orientation of the type j target in the cell. In an actual application scenario, the value of β^i_j would be obtained from available data on the effectiveness of each munition on various types of targets.

The attack capability $\xi^A_{ij}(t)$ is defined as

$$\xi^A_{ij}(t)=\begin{cases}\bar{\beta}^i_j\left(\mu^i(t)\right), & \text{if } \bar{\beta}^i_j\left(\mu^i(t)\right)\geq\xi_{\min}\\ 0, & \text{otherwise}\end{cases}$$

where

$$\bar{\beta}^i_j\left(\mu^i(t)\right)=\max_{\theta\in[0,2\pi)}\beta^i_j\left(\mu^i(t)\right)$$

$\bar{\beta}^i_j(\mu^i(t))$ represents the UAV's best attacking expertise achievable using the munition resources $\mu^i(t)$. If $\bar{\beta}^i_j(\mu^i(t))$ is less than some positive number $\xi_{\min}$, the UAV is not eligible to attack a type j target and $\xi^A_{ij}(t)=0$.

IV. Target Information Dynamics

A. Target Orientation Dynamics

Since the effectiveness of both sensor observations and attack depend on the orientation of a UAV relative to the target, it is important for UAVs to estimate the actual orientation of each observed target. Initially, UAVs have a default value of the actual orientation, and it is updated by observations. When UAV u_i takes a sensor reading at time t, observations $b_i(X,Y,t)\in\{0,1,2,\ldots,N_T\}$ are produced for all cells (X,Y) in its sensor field at the time. $b_i(X,Y,t)$ is a random variable depending on the sensing capability and observation angle, 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 no target detection). If this UAV makes the observation 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

$$\Phi_j(X,Y,t)=\left[1-\gamma^i_j\left(\theta_S(t),\zeta^i(t)\right)\right]\Phi_j(X,Y,t^-)+\gamma^i_j\left(\theta_S(t),\zeta^i(t)\right)\phi_t \quad (1)$$

where t^- indicates the time immediately prior to the observation $\theta_S(t)$ is the RAO given by the angle between the UAV's heading and the target's estimated orientation, and $\gamma^i_j(\theta_S(t),\zeta^i(t))\in[0,1]$ is the capability of the sensor resource $\zeta^i(t)$ for estimating the orientation of type j targets from an RAO of $\theta_S(t)$. For an actual UAV, the orientation sensing capability γ^i_j would be obtained from known characteristic data for the sensors carried by the UAV. Note that the target orientation is updated only for the target type detected at time t. The UAVs' estimate orientation eventually approaches the true value after several observations.

$\Phi_j(X,Y,t)$ is an estimate of the actual target orientation from the viewpoint of UAVs. The confidence can be measured by introducing a variable that represents how certain the team is about the estimate orientation (similar to the uncertainty about target existence and target type discussed later in Section IV-C).

B. TOP Dynamics

The TOP map is updated in an event-driven fashion by UAVs' observations and actions. When UAV u_i takes a sensor reading at time t, observations $b_i(X,Y,t)$ are produced for all cells (X,Y) in its sensor field at the time. If 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

$$P(X,Y,t) = F\left(P(X,Y,t^-), T(X,Y,t^-), a(X,Y,t), b(X,Y,t)\right) \quad (2)$$

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

The TOP update function F is defined for the cases of observation and action.

Observation-Triggered TOP Update Function F_o: If UAV u_i makes a sensor reading $b_i(X,Y,t) \in \{0,1,2,\ldots,N_T\}$ in cell (X,Y) at time t using sensor resources $\zeta^i(t)$, the TOP for each target type, $j = 0,1,2,\ldots,N_T$, is updated based on the Bayesian formulation given as

$$P_j(X,Y,t) = \frac{\lambda^i_{j,b_i(X,Y,t)}\left(\theta_S(t),\zeta^i(t)\right) P_j(X,Y,t^-)}{\sum_{l=0}^{N_T} \lambda^i_{l,b_i(X,Y,t)}\left(\theta_S(t),\zeta^i(t)\right) P_l(X,Y,t^-)} \quad (3)$$

where $P_j(X,Y,t^-)$ is the TOP prior to the observation. Note that the TOP for all target types in the cell are updated after each observation, and $\sum_{j=0}^{N_T} P_j(X,Y,t) = 1$ at all times. The Bayesian update rule is widely used in sensor literatures [72]–[75]. The TOP update equation is discussed in greater detail in Appendix A [69], [76].

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

$$P_j(X,Y,t) = \left[1 - \beta^i_j\left(\theta_A(t),\mu^i(t)\right)\right] P_j(X,Y,t^-) \quad \text{for } j = 1,2,\ldots,N_T \quad (4)$$

$$P_0(X,Y,t) = 1 - \sum_{l=1}^{N_T} P_l(X,Y,t) \quad (5)$$

where $P_j(X,Y,t^-)$ is the TOP for target type j prior to attack, and by the definition of $\beta^i_j(\theta_A(t),\mu^i(t))$, $1-\beta^i_j(\theta_A(t),\mu^i(t))$ denotes the probability that attack with munition $\mu^i(t)$ from RAA $\theta_A(t)$ does not destroy a type j target. As with the observation-triggered updates, the TOP for all target types is updated after an action.

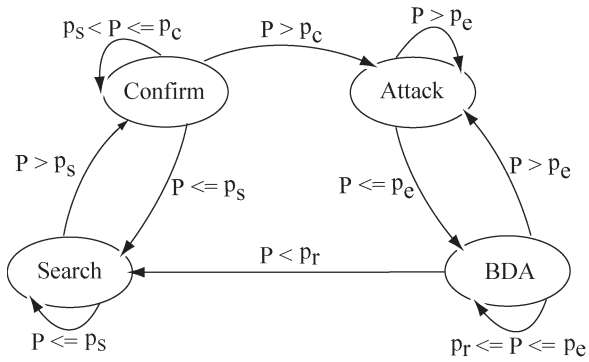

Fig. 2. Automaton formulation of the task dynamics, where p_s is the suspicion threshold, p_c is the certainty threshold, p_e is the exit threshold, and p_r is the resolution threshold.

C. Uncertainty Dynamics

In order to direct the search for targets efficiently, it is important to quantify how much is known about the existence of a live target in each cell as well as the target type. This is done by defining an uncertainty variable for each cell and updating it as the cell is observed or a target in it is attacked. We find it useful to distinguish between two types of uncertainty, namely 1) uncertainty about the existence of a target within a cell and 2) uncertainty about the type of the target, if any. We use an entropy-based formulation for each, defining the two quantities as follows:

1) Target occupancy uncertainty $\chi_o(X,Y,t)$, given by

$$\chi_o(X,Y,t) = -P_0(X,Y,t)\log P_0(X,Y,t) - (1-P_0(X,Y,t))\log\left(1-P_0(X,Y,t)\right).$$

Since statistical entropy is a probabilistic measure of uncertainty or ignorance, $\chi_o(X,Y,t)$ indicates how uncertain the UAV team is about the existence of a live target in cell (X,Y) at time t. It is maximal when the presence or absence of a target are equiprobable and zero when one or the other is certain.

2) Target type uncertainty $\chi_t(X,Y,t)$, given by

$$\chi_t(X,Y,t) = -\frac{1}{\log N_T}\sum_{l=1}^{N_T}\left(P_l(X,Y,t)\log P_l(X,Y,t)\right)$$

where $\chi_t(X,Y,t)$ quantifies how uncertain the UAV team is about the type of a live target in cell (X,Y) at time t.

Based on the above definitions, an uncertainty variable $\chi(X,Y,t)$ for each cell (X,Y) is defined as

$$\chi(X,Y,t) = \omega_\chi \chi_o(X,Y,t) + (1-\omega_\chi)\chi_t(X,Y,t) \quad (6)$$

where ω_χ is the uncertainty coefficient that takes a value between 0 and 1. This combined entropy-like formulation provides a measure that represents how uncertain the UAV team is that a target exists in (X,Y) and about its type. It should be noted that uncertainty here is a subjective quantity based on the UAVs' estimate of what they know, not an objective measure of what is known. This kind of measure is also used in our previous work to guide UAVs' search path [38].

D. Task Dynamics

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

$$T(X,Y,t) = H\big(T(X,Y,t^-), P(X,Y,t); \bar{\rho}\big) \tag{7}$$

where the parameter vector $\bar{\rho}$ represents the set of threshold values used for transitions. The dynamics is made stochastic by the random nature of $b(X,Y,t)$ and the TOP update function F.

Fig. 2 shows the transitions between states using an automaton formulation. The task update function H is defined separately for each task status.

Task 1—Search: A cell (X,Y) with the search status for targets of type j transitions to confirm for that target type if $P_j(X,Y,t)$ exceeds the suspicion threshold p_s; otherwise, it remains in the search status.

Task 2—Confirm: The cuing of a confirm task for a target of type j at cell (X,Y) indicates that a UAV with the appropriate sensors should move towards the cell and make an observation in it. Unlike search, the confirm task is assignable to UAVs with the appropriate expertise. The sensors used may also be different in the two cases. The execution of the task causes an observation-triggered TOP update. The cell transitions to search for the type j target if $P_j(X,Y,t)$ falls below p_s (as a result of failure to confirm suspicions) and to attack for the type j target if $P_j(X,Y,t)$ exceeds the certainty threshold p_c (i.e., the existence of the type j target is confirmed from the UAVs' viewpoint).

Task 3—Attack: The attack status for a target of type j indicates that an appropriately armed UAV should proceed to the location and attack the target there with the correct munition. The execution of the task causes an action-triggered TOP update. If the updated $P_j(X,Y,t)$ falls below the exit threshold p_e (as a result of successful attack), the cell transitions to status BDA for the type j target; otherwise, it remains in the attack status.

Task 4—BDA: The purpose of a BDA task for target type j is to verify that, after an attack in cell (X,Y), the TOP for target type j has indeed fallen below p_e. Like confirm, the task is assigned to a UAV with the appropriate sensors, which proceeds to (X,Y) and makes an observation, leading to an observation-triggered update. If the result of the update produces $P_j(X,Y,t) \geq p_e$ (i.e., the target was not eliminated from the UAVs' viewpoint), the task status for target type j transitions back to attack; if $p_r \leq P_j(X,Y,t) \leq p_e$ (i.e., the UAVs are not sure whether the target was eliminated or not), it remains BDA; and if $P_j(X,Y,t) < p_r$ (i.e., the target is neutralized), it transitions to search.

The task dynamic model is built from the UAVs' viewpoint, so it is reasonable to assume that UAVs know the transition thresholds. These values can be chosen based on the particular application and requirements. For example, if the mission area is dangerous and missing, a target is not tolerated, the suspicion threshold p_s, certainty threshold p_c, and resolution threshold p_r could be set low, while the exit threshold could be set high.

V. Assignment Algorithms

The primary focus of our work is to develop an efficient algorithm that assigns UAVs to tasks at target locations such that the mission is completed as rapidly and cheaply as possible. The problem is complicated by three factors.

1) Not all target locations and types are known *a priori*.
2) The task transitions even at known target locations are stochastic.
3) The effectiveness of each UAV depends on task and target type, so all UAVs cannot be considered equivalent.

In order to obtain flexibility and scalability, the basic algorithm that we propose comprises an initial assignment of at most one target to each UAV followed by an in-field event-driven process where UAVs select tasks as they become available using a policy of gradual commitment. In this approach, UAVs do not attempt to predict future tasks. This issue is addressed in our predictive algorithm, where UAVs use their current information and problem model to estimate probabilities for future tasks and incorporate these predictions into their decision making. In principle, this allows for better coordination and faster response. However, there are certain hazards associated with prediction, and consideration of the costs and benefits of prediction is a major focus of this paper.

A. Basic Assignment Algorithm

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

It is assumed that all UAVs have instantaneous and noise-free access to a centralized information base (IB), which comprises the following items:

1) the task status map $T(X,Y,t)\ \forall (X,Y)$;
2) the assignment status map $A(X,Y,t)\ \forall (X,Y) \in L$;
3) the orientation map $\Phi(X,Y,t)\ \forall (X,Y)$;
4) the TOP map $P(X,Y,t)\ \forall (X,Y)$;
5) the uncertainty map $\chi(X,Y,t)\ \forall (X,Y)$;
6) the UAV state vector $S(t) = \{S_i(t)\}\ \forall u_i$.

The IB is updated in an event-driven manner whenever new information comes up.

1) Status Initialization: Initially, if a cell (X,Y) is suspected to have a target of type j^*, the probability $P_j(X,Y,0)$ for $j = 0,1,2,\ldots,N_T$ is assigned as

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

i.e., the suspected target type is given a probability higher than the threshold necessary for the confirm, and the remaining

probability is divided equally among the other possibilities. This leads to the initial task status being set as

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

For those cells that have no suspected target of any type, the UAV team has no reason to assume that a target of some type is more probable than the others. Thus, it is natural to set $P_j(X,Y,0)$, for $j = 0,1,2,\ldots,N_T$ equiprobable as

$$P_j(X,Y,0) = \frac{1}{1+N_T}, \qquad \text{for } j = 0,1,2,\ldots,N_T$$

and

$$T_j(X,Y,0) = \text{search}, \qquad \text{for } j = 0,1,2,\ldots,N_T.$$

The orientation estimate $\Phi_j(X,Y,0)$ begins with 0 for all (X,Y) and $j = 0,1,2,\ldots,N_T$. The UAVs' initial positions are randomly distributed in the environment. All UAVs initially have the open status.

2) Initial Assignment: 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 as

$$h_{ik} = \omega_c \bar{d}_{ik} + (1-\omega_c)\exp(-\xi_{i,m_k}) \tag{8}$$

where ω_c is the cost coefficient that takes a value between 0 and 1, d_{ik} is the distance between UAV u_i and the location of task τ_k, the normalized distance $\bar{d}_{ik} = \max_{i,k} d_{ik}$, and ξ_{i,m_k} is the capability of UAV u_i for task m_k. UAV u_i is eligible for task m_k if the capability $\xi_{i,m_k} \geq \xi_{\min}$, where $\xi_{\min}$ is a nonnegative parameter.

Each UAV reports its cost for all tasks in the CTL for which it is eligible to the central controller, which then uses a bipartite matching algorithm [77] (or linear programming [78]) to match UAVs with tasks to minimize the total cost in the team. The central controller is off-line and only does the initial assignment. UAVs that are within distance D_a (termed the commitment threshold) of their matched tasks are assigned to the task and are given the committed status, while UAVs that are farther away are associated with their matched tasks and are given the competing status. It is assumed that only one UAV is allowed to be assigned to a task, but up to n_a UAVs can be associated with one task. Similarly, each UAV can only be committed to a single task, but it can be competing for up to m_a tasks. UAVs assigned to a task determine the best RAO (or RAA) with respect to their sensors (or munitions) and plan an optimal path to approach the target from that angle [79]. When a UAV has no task to choose (this can happen when all the targets are neutralized or the UAV is ineligible for all available or associated assignable tasks), it has an open status and follows a path of maximum local uncertainty. An open

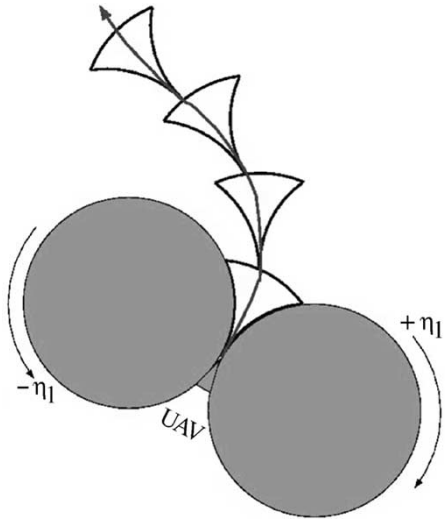

Fig. 3. Search path. The fan-shaped polygons denote the UAV's flyable area subject to the turning constraints.

UAV calculates its flyable area over a finite horizon based on its speed, heading angle, and turning constraints (denoted by the fan-shaped regions in Fig. 3). The UAV considers $n_p + 1$ angularly equally spaced paths through this region, with the extreme paths corresponding to the boundaries of the turning circles. For each path, the UAV calculates the total uncertainty of the cells included in it and chooses the path with the maximum total uncertainty. As it approaches the end of the chosen path segment, the UAV repeats the process over the next finite horizon region. Thus, the UAV follows a path of maximum local uncertainty, much as a steepest descent algorithm follows a path of locally steepest descent. The purpose is to maximize the benefit from search in a greedy way, and the path that followed is termed as a search path. Increasing the horizon considered at each step can improve the path but becomes computationally prohibitive.

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 the search paths. All UAVs take sensor readings as they move (as described earlier) and update the TOP in the cells scanned. 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.

3) Assignment Update: At all times, all open and competing UAVs monitor the CTL (which is being updated continually) 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 disassociated (or released) from it. At any time, as many as n_a open or competing UAVs can be associated with it, while no UAV can be associated with more than m_a different

tasks. Thus, if a better UAV becomes available for a task, another UAV associated with it may be released. Sometimes, a task may be done opportunistically by a UAV that happens to pass by, in which case, all UAVs associated with or assigned to it are released. At all times, all committed UAVs move to their assigned task, UAVs associated with multiple tasks move towards the one with the lowest cost, and open UAVs follow a search path. The process continues until all the targets are neutralized and the uncertainty of all locations has fallen below some positive threshold or some time limit is reached.

B. Predictive Assignment Algorithm

The basic 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 including 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 position and heading angle are denoted as $(x_{i^*}(t), y_{i^*}(t))$ and $\psi_{i^*}(t)$, respectively, and $\theta_{i^*}(t)$ is its planned approach angle for the current task. Given the above information, 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. Furthermore, using the update (3)–(5), 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 volunteering for the predicted tasks according to their distance from the task location and capabilities. The commitment procedure for the predicted tasks is the same as for the current ones. Consider a UAV u_i with location $(x_i(t), y_i(t))$, heading angle $\psi_i(t)$, and sensing (or attacking) capability 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

$$\delta_{i,(X,Y)} = (t^* - t) - t_{i,(X,Y)} \tag{9}$$

where t denotes the current time. 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.

A function $Q(z)$ is defined 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

$$h_{ik}^P = \omega_c Q\left(\overline{\delta}_{i,(X,Y)}\right) + (1 - \omega_c) \exp\left(-\pi(X, Y, t^*) \cdot \xi_i(t)\right) \tag{10}$$

where $\overline{\delta}_{i,(X,Y)}$ is the normalized time cushion value defined as $\overline{\delta}_{i,(X,Y)} = \delta_{i,(X,Y)} / \max_i |\delta_{i,(X,Y)}|$.

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.

C. Performance Measures

The goal for the UAV team is to cover the environment as rapidly as possible in such a way that the whole region is completely searched and all targets neutralized. Specially, two time measures are defined to qualify performance.

1) Target neutralization time (TNT), which is the time needed to actually neutralize all targets. This is an objective measure of performance that is not based on the UAVs' estimation of TOPs.
2) Total mission time (TMT), which is the total time needed to destroy all targets and bring the uncertainty of all locations below some positive threshold. Unlike TNT, the TMT is a subjective measure from the UAVs' viewpoint.

Based on the developed model formulation and cooperative search and task assignment algorithms, in this paper, we investigate the intrinsic value of prediction as well as the search–response tradeoff.

D. Intrinsic Value of Prediction

The main benefit of using predictive assignment is better coordination among the UAVs and better response on the assignable tasks. However, it also has some drawbacks, namely 1) computational costs; 2) wasted effort caused by incorrect prediction; and 3) loss of search resources. The degree to which the benefits of prediction can be realized in the face of its costs may also depend on the size of the UAV team relative to the mission at hand. If the available number of UAVs is large relative to the number of targets, the cost of prediction can be absorbed without affecting performance, where the benefits can still be realized. The proposed model is applied to study intrinsic value of prediction as a function of UAV team composition.

E. Search–Response Tradeoff

In the model described in the previous sections, 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 tradeoff [80], where resources

dedicated to search (exploration) compete with those dedicated to target response (exploitation). This is termed the search–response tradeoff. This tradeoff 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. The search–response tradeoff is studied systematically using the proposed model and two hypotheses.

1) Prediction may help when (or once) most targets are known, since quick response is more important for currently known targets.
2) Prediction could hurt when most targets are unknown, since efficient search is necessary to find unknown targets.

We try to answer two questions based on the results from Monte Carlo simulations.

1) Is there a crossover in the performance of the predictive and nonpredictive algorithms as the degree of target knowledge varies? If so, how does it depend on environment size, UAV team composition, number of targets, and other factors?
2) How does the benefit of prediction over nonprediction depend on environment size and number of targets relative to the number of UAVs?

VI. Simulation Results and Discussion

A. Simulation Scenario

To evaluate the performance of the basic and predictive assignment algorithms, Monte Carlo simulations are conducted using an event-driven simulator. In the simulations, there are two types of targets (Type 1 and Type 2), which are characterized by differences in their orientation angles. The UAVs are drawn from two classes, namely 1) target recognition (TR) UAVs and 2) attack (A) UAVs. All UAVs are assumed to have sensors needed for search but with different sensing capabilities. The sensing capabilities of UAVs from each class are defined using functions $\lambda^C_{j,k}$; the probability that a UAV of type C detects a type k target given that the true target type is j. The λ functions are chosen to reflect the fact that target identification is dependent on the angle of observation and the sensor resources used. We have used phenomenologically reasonable definitions of these functions for the purpose of simulation (e.g., giving the lowest probability of error at the optimal angle of observation, and lower probability of identification error for TR UAVs compared to A UAVs). In practice, the λ functions will be derived from a knowledge of the sensors, automatic target recognition (ATR) systems, and targets. The same considerations apply to the β^i_j functions for the case of attack. Please refer to Appendix B for the detailed list of design parameters.

B. Simulation Results

Since the UAVs move at a constant speed, a time unit is defined as the time needed to travel some distance. The time in the following simulations is measured by this time unit.

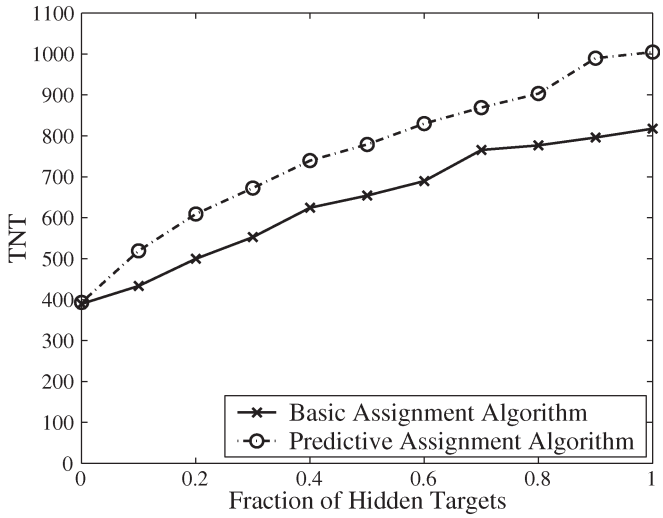

Fig. 4. TNT versus hidden target fraction: 150 by 150 km environment, ten targets, two TR UAVs, and two A UAVs.

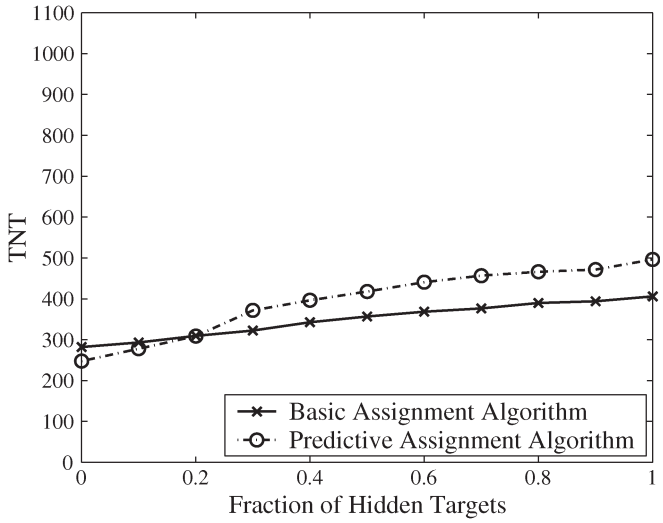

Fig. 5. TNT versus hidden target fraction: 150 by 150 km environment, ten targets, five TR UAVs, and five A UAVs.

The algorithms are coded in MATLAB, and the following simulation results are the average of 100 runs.

Case Study 1: The first set of simulation results (Figs. 4–9) compares the basic and predictive algorithms in terms of TNT for a UAV team with different composition in a 150 by 150 km environment with ten stationary targets. The number of targets whose positions are suspected *a priori* is varied from ten (no hidden targets) to 0 (all targets hidden).

In the first simulation (Fig. 4), the UAV team comprises two TR and two A UAVs. Observe that the basic algorithm is almost always better than the predictive algorithm, indicating that prediction does not provide any advantage even when all targets are known *a priori*. Intuitively, this is because, given the high number of targets relative to the UAVs, most UAVs are too busy with current tasks to take advantage of prediction. This suggests the existence of what might be termed "room for

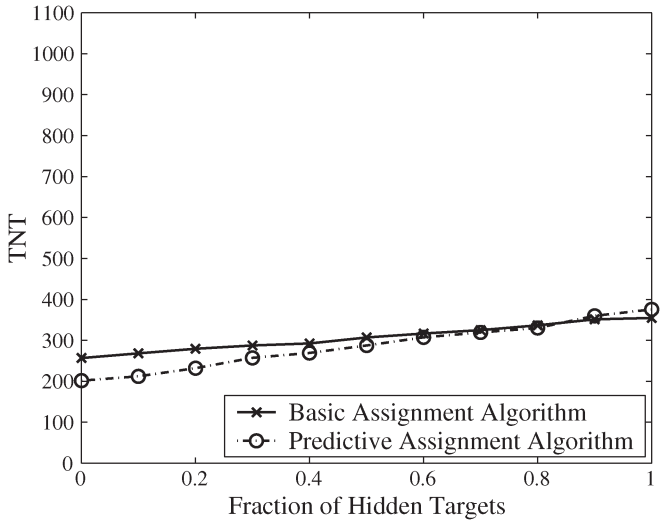

Fig. 6. TNT versus hidden target fraction: 150 by 150 km environment, ten targets, eight TR UAVs, and eight A UAVs.

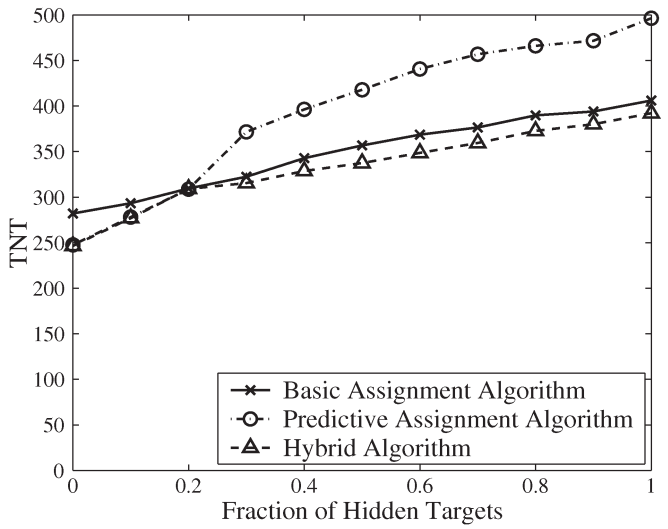

Fig. 7. TNT versus hidden target fraction: 150 by 150 km environment, ten targets, five TR UAVs, and five A UAVs.

prediction," i.e., how much opportunity the UAV team has to exploit the benefits of prediction even if it would help.

In the second simulation (Fig. 5), the UAV team consists of five TR and five A UAVs. The graph clearly shows the crossover predicted by the hypothesized search–response tradeoff. 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 nonpredictive one.

In the last simulation (Fig. 6), the UAV team is comprised of eight TR and eight A UAVs. Thus, the number of UAVs is higher relative to the number of targets than in the situations shown above. This time, prediction provides a significant advantage until the number of hidden targets reaches eight out of ten. This is consistent with the 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 the predicted tasks, thus, reducing the overall TNT. However, this does not neutralize the essential search–response tradeoff, and when almost all targets are hidden, the loss of search resources due to predictive assignment does become a liability.

Case Study 2: Having ascertained that the value of prediction depends on how informed the UAV team is about targets, we considered a hybrid algorithm, which works as follows.

When prior knowledge of targets is not high (so that search is needed), UAVs start out in the nonpredictive mode but switch to the predictive algorithm once a sufficient number of targets have been found. The simulation result (Fig. 7) shows that this algorithm works better than the predictive and the nonpredictive approaches and captures the best of both. 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 how certain it is about the environment relative to the available resources and use this as the basis of switching between predictive and nonpredictive assignment.

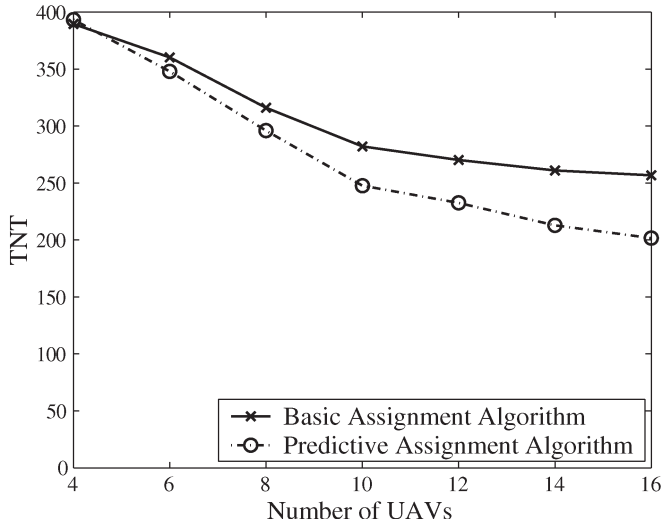

Fig. 8. TNT versus number of UAVs: 150 by 150 km environment and ten suspected targets.

Case Study 3: The third set of simulation results (Figs. 8 and 9) compares the basic and predictive algorithms for UAV teams of different sizes in a 150 by 150 km environment with ten suspected targets and no hidden targets. Fig. 8 shows that the benefit of predictive algorithm becomes larger as the size of the UAV team increases. This confirms our earlier hypothesis that a larger team is able to exploit prediction more effectively than a small team that nearly always is fully engaged with the current tasks. Fig. 9 shows that the predictive approach always

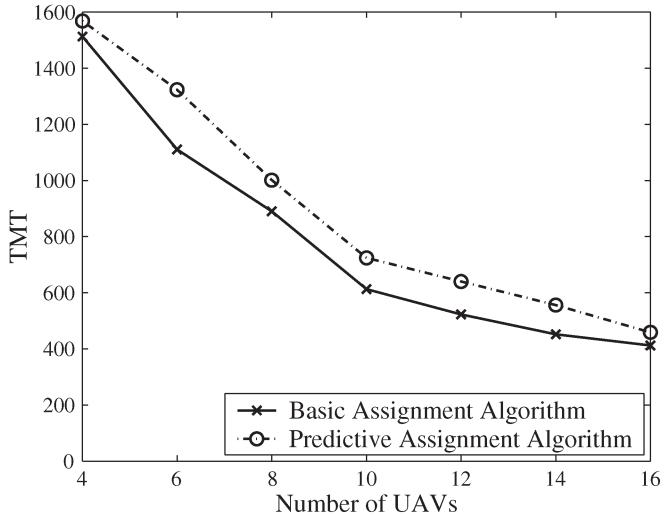

Fig. 9. TMT versus number of UAVs: 150 by 150 km environment and ten suspected targets.

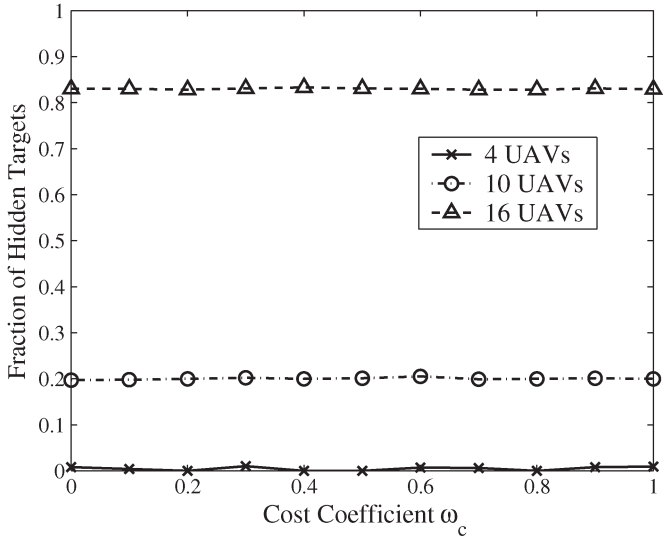

Fig. 11. Crossover point: 150 by 150 km environment and ten targets.

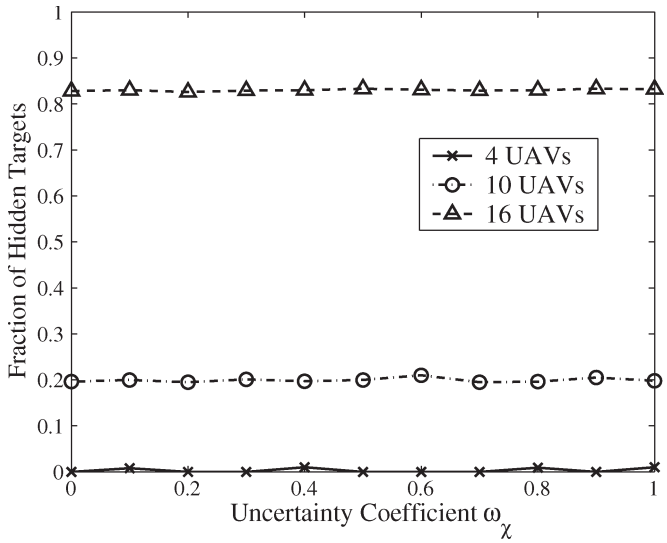

Fig. 10. Crossover point: 150 by 150 km environment and ten targets.

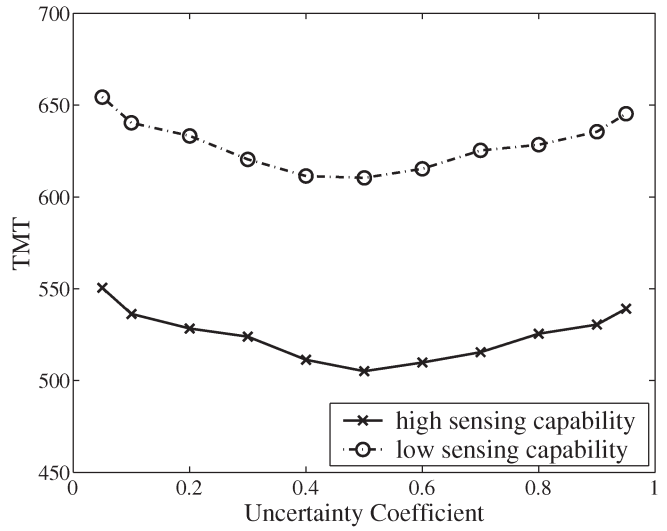

Fig. 12. Uncertainty coefficient effect on TMT: 150 by 150 km environment, ten suspected targets, and ten TR UAVs.

takes longer to neutralize all the targets and to complete the search, since even in the best case, going for predicted targets takes resources away from search.

Case Study 4: The fourth set of simulation results (Figs. 10 and 11) investigates the impact of two system design parameters, i.e., uncertainty coefficient ω_χ and cost coefficient ω_c, on the crossover point of the basic and predictive algorithms. The mission size is 150 by 150 km environment with ten targets. We consider three typical UAV team sizes in each case. The results show that the crossover point is insensitive to the change of these two parameters.

Case Study 5: This set of simulation (Figs. 12 and 13) considers the impact of the certainty coefficient ω_χ in (6) on the performance of the UAV team. As ω_χ increases, the uncertainty value used to guide search is weighted more strongly towards uncertainty about the existence of targets and less toward uncertainty about target type. Operationally, this means that, at high values of ω_χ, searching for targets in new locations will be preferred to spending effort verifying the type of already discovered targets. This, in turn, is expected to affect overall performance, since targets differ in optimal observation angles according to type, and targets of uncertain type are likelier to be observed (and later attacked) from the wrong direction, possibly by the wrong type of UAV. Since ω_χ is primarily a driver for search, the simulations here focus only on the search and identification tasks using only TR UAVs. It should be noted, however, that in a full mission, efficiency of search would also affect the time to neutralize targets. The UAV team comprises ten TR UAVs in a 150 by 150 km environment with ten initially unknown targets (five of each type). The mission is to do a complete search and to classify all the targets. The locations of UAVs and targets are generated randomly in each run. Each data point is averaged over 200 independent runs. We do two separate sets of simulations: one where UAVs have a

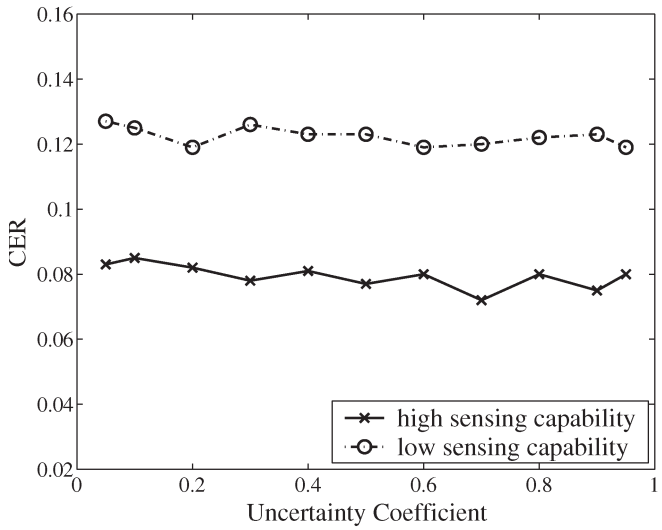

Fig. 13. Uncertainty coefficient effect on CER: 150 by 150 km environment, ten suspected targets, and ten TR UAVs.

high sensing capability (the probability of correct classification is 0.98 from optimal observation angles and 0.7 from other angles) and another where it is lower (the probability of correct classification is 0.9 from optimal observation angles and 0.6 from other angles).

Fig. 12 shows that the total time for complete search has a shallow U-shaped dependence on ω_χ. The value of the optimal uncertainty coefficient appears to depend slightly on sensing capability but is close to $\omega_\chi = 0.5$, indicating that equal weight to target presence and type is a good policy. As expected, UAVs with higher sensing capability finish faster.

Fig. 13 plots the effect on classification error rate (CER), which is defined as the fraction of targets missed or classified incorrectly. This is an objective measure of search quality, addressing the possibility that the UAVs' assessment may differ from reality. The figure shows that the CER is virtually independent of ω_χ, though, as expected, UAVs with lower sensing capability make more errors. Thus, the overall conclusion is that, given the sensing capability of the UAV team, ω_χ can be chosen to optimize TMT without concern for its effect on classification error.

VII. Conclusion

In this paper, we have presented an approach for the allocation of tasks to UAVs' involved in a search-and-act mission. Three cooperative algorithms have been described for heterogeneous UAV teams in a spatially extended battlefield environment with stationary targets. It has been shown that prediction can help improve the performance of the cooperative UAV teams, but 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. The simulations also show that there is a well-defined crossover point as this knowledge changes. The results show that the crossover point is almost independent of the system design parameters ω_χ and ω_c.

While the work we have presented focuses on UAV teams, it is broadly applicable to teams of mobile agents acting cooperatively in an extended environment in applications such as surveillance, search and rescue, wildfire monitoring, etc. The broader contributions of this work are stated as follows: 1) description of a simple easily decentralizable approach to multiagent spatial task assignment in dynamic environments where task dynamics is coupled with agent behavior; 2) explicit elucidation of a tradeoff between prediction-based "exploitation" and search-based "exploration" for agent teams involved in search-and-act missions; and 3) a demonstration that the utility of prediction depends on (at least) two factors, namely a) the "space for prediction" created by the size of the agent team relative to the number of task locations and b) the knowledge that the agent team has about the tasks. These results suggest that an intelligent adaptive use of prediction could be valuable in spatial multiagent problems.

Appendix A
Derivation of (3)

To obtain the update function (3), consider the case where a UAV takes a measurement in cell (X, Y) at time t. Define the following for a cell (X, Y).

1) E_j is the event that a target of type j is actually located in cell (X, Y).
2) $b(X, Y, t)$ is the sensor reading taken by the UAV, $b(X, Y, t) \in \{0, 1, 2, \ldots, N_T\}$ with $b(X, Y, t) = 0$, indicating no target detection.
3) $B(X, Y, t^-)$ is the vector of all sensor readings for cell (X, Y) by all UAVs taken before time t.
4) $P(E_j|B(X, Y, t^-))$ is the probability that a target of type j is in cell (X, Y) based on observations prior to the current one.
5) $P(E_j|B(X, Y, t^-), b(X, Y, t))$ is the updated probability after obtaining the new reading, $b(X, Y, t)$.

It is assumed that the sensors' measurements in any cell are conditionally independent given the state of the cell. That is

$$P\left(b(X,Y,t)|B(X,Y,t^-), E_j\right) = P\left(b(X,Y,t)|E_j\right). \quad (11)$$

This is a common assumption in the sensor fusion literature [76].

Based on the above definitions and assumptions, the updating function (3) follows directly from Bayes' rule [76], as in (12), shown at the bottom of the next page, which can be simplified by applying the conditional independence assumption [see (11)] written as

$$P\left(E_j|B(X,Y,t^-), b(X,Y,t)\right)$$

$$= \frac{P\left(b(X,Y,t)|E_j\right) P\left(E_j|B(X,Y,t^-)\right)}{\sum_{l=0}^{N_T} P\left(b(X,Y,t)|E_l\right) P\left(E_l|B(X,Y,t^-)\right)}. \quad (13)$$

The TOP $P_j(X, Y, t)$, which is calculated by the model, is an estimate of $P(E_j|B(X, Y, t)$, i.e., the estimated probability

that the cell (X, Y) contains a live target of type j, $j = 0, 1, 2, \ldots, N_T$, with $j = 0$ denoting the no target case. Thus, (13) can be written as

$$P_j(X,Y,t) = \frac{P\left(b(X,Y,t)|E_j\right) P_j(X,Y,t^-)}{\sum_{l=0}^{N_T} P\left(b(X,Y,t)|E_l\right) P_l(X,Y,t^-)}. \quad (14)$$

The probabilities $P(b(X,Y,t)|E_l)$ are characteristics of the sensor resources used and are assumed to be given by the sensor specification. The capabilities of a sensor resource suite $\zeta(t)$ are specified by $\lambda_{j,b(X,Y,t)}(\theta_S(t), \zeta(t)) = P(b(X,Y,t)|E_j; \theta_S(t), \zeta(t))$, where $\theta_S(t)$ is the observation angle. Thus, for an observation made from angle θ_S, (14) gives the update (3) as

$$P_j(X,Y,t) = \frac{\lambda_{j,b(X,Y,t)}\left(\theta_S(t), \zeta(t)\right) P_j(X,Y,t^-)}{\sum_{l=0}^{N_T} \lambda_{l,b(X,Y,t)}\left(\theta_S(t), \zeta(t)\right) P_l(X,Y,t^-)}.$$

It should be noted that this update equation uses subjective estimates of probabilities and observation angles, and, therefore, includes some uncertainty. These uncertainties can be reduced by incorporating accurate prior information and by using better sensors. This issue is not the focus of the present work.

Appendix B
Simulator Design Parameters

The simulations require the selection of several parameters and the specification of ATR and munition efficiency models. The parameters were chosen as follows:

1) cost parameter $\omega_c = 0.5$;
2) uncertainty parameter $\omega_\chi = 0.5$;
3) task transition thresholds: resolution threshold $p_r = 0.03$, suspicion threshold $p_s = 0.55$, existence threshold $p_e = 0.7$, and certainty threshold $p_c = 0.9$.

These values should be seen as logically reasonable choices rather than specific recommendations. Other values might be appropriate for specific mission and circumstances. In particular, the choice of task transition thresholds will depend on the type of mission. For example, the threshold for attacking buildings may be higher than that for attacking antiaircraft gun position.

For the ATR model, we again use a logically reasonable phenomenological model (described below). In practice, the model used for a specific UAV team will depend on the actual sensor resources available and their known characteristics. Any model where sensor capabilities can be specified using the very general framework we propose (i.e., using λ parameters) is compatible with our overall approach.

The sensor accuracy parameter $\lambda^C_{j,k}(\theta_S(t), \zeta^i(t))$ is shown at the top of the next page.

The munition efficiency model is also defined phenomenologically, using a reasonable function. The munition capability parameter $\beta^C_j(\theta_A(t), \mu^C(t))$ is defined as

$$\beta^{\mathrm{TR}}_j\left(\theta_A(t), \mu^{\mathrm{TR}}(t)\right) = 0 \qquad \text{for } j = 1, 2$$

$$\beta^A_1\left(\theta_A(t), \mu^A(t)\right) = 0.95\frac{2 + \cos\left(\theta_A(t) - \frac{3}{4}\pi\right)}{3}$$

$$\beta^A_2\left(\theta_A(t), \mu^A(t)\right) = 0.95\frac{2 + \cos\left(\theta_A(t) + \frac{3}{4}\pi\right)}{3}.$$

Acknowledgment

The authors thank P. Chandler, A. Sparks, S. Rasmussen, C. Schumacher, M. Mears, K. Passino, R. Ordóñez, and M. Flint for the helpful suggestions and acknowledge access to the AFRL/VACA MultiUAV Simulator v 1.1 trajectory generator provided by S. Rasmussen. The authors also thank the Associate Editor and all referees for their helpful suggestions and input.

$$\begin{aligned}
P\left(E_j|B(X,Y,t^-), b(X,Y,t)\right) &= \frac{P\left(E_j, B(X,Y,t^-), b(X,Y,t)\right)}{P\left(B(X,Y,t^-), b(X,Y,t)\right)} \\
&= \frac{P\left(b(X,Y,t)|B(X,Y,t^-), E_j\right) P\left(B(X,Y,t^-), E_j\right)}{P\left(B(X,Y,t^-)\right) P\left(b(X,Y,t)|B(X,Y,t^-)\right)} \\
&= \frac{P\left(b(X,Y,t)|B(X,Y,t^-), E_j\right) P\left(E_j|B(X,Y,t^-)\right) P\left(B(X,Y,t^-)\right)}{P\left(B(X,Y,t^-)\right) P\left(b(X,Y,t)|B(X,Y,t^-)\right)} \\
&= \frac{P\left(b(X,Y,t)|B(X,Y,t^-), E_j\right) P\left(E_j|B(X,Y,t^-)\right)}{P\left(b(X,Y,t)|B(X,Y,t^-)\right)} \\
&= \frac{P\left(b(X,Y,t)|B(X,Y,t^-), E_j\right) P\left(E_j|B(X,Y,t^-)\right)}{\sum_{l=0}^{N_T} P\left(b(X,Y,t)|B(X,Y,t^-), E_l\right) P\left(E_l|B(X,Y,t^-)\right)}
\end{aligned} \quad (12)$$

$$\lambda_{0,0}^{\mathrm{TR}}\left(\theta_S(t),\zeta^{\mathrm{TR}}(t)\right)=0.95$$

$$\lambda_{1,1}^{\mathrm{TR}}\left(\theta_S(t),\zeta^{\mathrm{TR}}(t)\right)=\begin{cases}0.95, & \text{if } \frac{5}{6}\pi\le\theta_S(t)\le\frac{7}{6}\pi\\ 0.6, & \text{if } \theta_S(t)\le\frac{2}{3}\pi \text{ or } \theta_S(t)\ge\frac{4}{3}\pi\\ 0.95+\frac{0.95-0.6}{\frac{\pi}{6}}\left(\theta_S(t)-\frac{5}{6}\pi\right), & \text{if } \frac{2}{3}\pi<\theta_S(t)<\frac{5}{6}\pi\\ 0.95-\frac{0.95-0.6}{\frac{\pi}{6}}\left(\theta_S(t)-\frac{7}{6}\pi\right), & \text{otherwise}\end{cases}$$

$$\lambda_{2,2}^{\mathrm{TR}}\left(\theta_S(t),\zeta^{\mathrm{TR}}(t)\right)=\begin{cases}0.95, & \text{if } \frac{4}{3}\pi\le\theta_S(t)\le\frac{5}{3}\pi\\ 0.6, & \text{if } \theta_S(t)\le\frac{7}{6}\pi \text{ or } \theta_S(t)\ge\frac{11}{6}\pi\\ 0.95+\frac{0.95-0.6}{\frac{\pi}{6}}\left(\theta_S(t)-\frac{4}{3}\pi\right), & \text{if } \frac{7}{6}\pi<\theta_S(t)<\frac{4}{3}\pi\\ 0.95-\frac{0.95-0.6}{\frac{\pi}{6}}\left(\theta_S(t)-\frac{5}{3}\pi\right), & \text{otherwise}\end{cases}$$

$$\lambda_{j,k}^{\mathrm{TR}}\left(\theta_S(t),\zeta^{\mathrm{TR}}(t)\right)=\frac{1-\lambda_{j,j}^{\mathrm{TR}}\left(\theta_S(t),\zeta^{\mathrm{TR}}(t)\right)}{2}\qquad \text{for } j\neq k,\ j=0,1,2,\ k=0,1,2$$

$$\lambda_{0,0}^{A}\left(\theta_S(t),\zeta^{A}(t)\right)=0.8$$

$$\lambda_{1,1}^{A}\left(\theta_S(t),\zeta^{A}(t)\right)=\begin{cases}0.8, & \text{if } \frac{5}{6}\pi\le\theta_S(t)\le\frac{7}{6}\pi\\ 0.5, & \text{if } \theta_S(t)\le\frac{2}{3}\pi \text{ or } \theta_S(t)\ge\frac{4}{3}\pi\\ 0.8+\frac{0.8-0.5}{\frac{\pi}{6}}\left(\theta_S(t)-\frac{5}{6}\pi\right), & \text{if } \frac{2}{3}\pi<\theta_S(t)<\frac{5}{6}\pi\\ 0.8-\frac{0.8-0.5}{\frac{\pi}{6}}\left(\theta_S(t)-\frac{7}{6}\pi\right), & \text{otherwise}\end{cases}$$

$$\lambda_{2,2}^{A}\left(\theta_S(t),\zeta^{A}(t)\right)=\begin{cases}0.8, & \text{if } \frac{4}{3}\pi\le\theta_S(t)\le\frac{5}{3}\pi\\ 0.5, & \text{if } \theta_S(t)\le\frac{7}{6}\pi \text{ or } \theta_S(t)\ge\frac{11}{6}\pi\\ 0.8+\frac{0.8-0.5}{\frac{\pi}{6}}\left(\theta_S(t)-\frac{4}{3}\pi\right), & \text{if } \frac{7}{6}\pi<\theta_S(t)<\frac{4}{3}\pi\\ 0.8-\frac{0.8-0.5}{\frac{\pi}{6}}\left(\theta_S(t)-\frac{5}{3}\pi\right), & \text{otherwise}\end{cases}$$

$$\lambda_{j,k}^{A}\left(\theta_S(t),\zeta^{A}(t)\right)=\frac{1-\lambda_{j,j}^{A}\left(\theta_S(t),\zeta^{A}(t)\right)}{2}\qquad \text{for } j\neq k,\ j=0,1,2,\ k=0,1,2$$

REFERENCES

[1] *Airborne Drones to Guide U.S. Forces in Afghanistan*. [Online]. Available:http://www.namibian.com.na/2001/September/world/0116DC7B3A.html

[2] A. Joulia and C. Le Tallec, "Multicriteria analysis tool for civil UAV configurations/applications matching," in *Proc. AIAA 3rd "Unmanned Unlimited" Technical Conf., Workshop and Exhibit*, Chicago, IL, Sep. 2004, pp. 90–100.

[3] M. Okrent, "Civil UAV activity within the framework of European Commission research," in *Proc. AIAA 3rd "Unmanned Unlimited" Technical Conf., Workshop and Exhibit*, Chicago, IL, Sep. 2004, pp. 101–111.

[4] S. Tsach, A. Peled, D. Penn, and D. Touitou, "The CAPECON program: Civil applications and economical effectivity of potential UAV configurations," in *Proc. AIAA 3rd "Unmanned Unlimited" Technical Conf., Workshop and Exhibit*, Chicago, IL, Sep. 2004, pp. 71–89.

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

[6] R. Beard, T. McLain, and M. Goodrich, "Coordinated target assignment and intercept for unmanned air vehicles," in *Proc. IEEE Int. Conf. Robotics and Automation*, Washington DC, May 2002, pp. 2581–2586.

[7] J. Bellingham, M. Tillerson, A. Richards, and J. How, "Multi-task allocation and path planning for cooperative UAVs," in *Cooperative Control: Models, Applications and Algorithms*, S. Butenko, R. Murphey, and P. Pardalos, Eds. Boston, MA: Kluwer, 2003, pp. 23–41.

[8] P. Chandler and M. Pachter, "Research issues in autonomous control of tactical UAVs," in *Proc. Amer. Control Conf.*, Philadelphia, PA, 1998, pp. 394–398.

[9] P. Chandler, M. Pachter, and S. Rasmussen, "UAV cooperative control," in *Proc. Amer. Control Conf.*, Arlington, VA, 2001, pp. 50–55.

[10] J. Finke, K. Passino, S. Ganapathy, and A. Sparks, "Modelling and analysis of cooperative control systems for uninhabited autonomous vehicles," in *Cooperative Control*, S. Morse, N. Leonard, and V. Kumar, Eds. New York: Springer-Verlag, 2004.

[11] D. Jacques, "Search, classification and attack decisions for cooperative wide area search munitions," in *Cooperative Control: Models, Applications and Algorithms*, S. Butenko, R. Murphey, and P. Pardalos, Eds. Boston, MA: Kluwer, 2003, pp. 75–93.

[12] S.-M. Li, J. Boskovic, S. Seereeram, R. Prasanth, R. Amin, R. Mehra, R. Beard, and T. W. McLain, "Autonomous hierarchical control of multiple unmanned combat air vehicles (UCAVs)," in *Proc. Amer. Control Conf.*, Anchorage, AK, 2002, pp. 274–279.

[13] T. McLain, R. Beard, and J. Kelsey, "Experimental demonstration of multiple robot cooperative target intercept," presented at the AIAA Guidance, Navigation, and Control Conf., Monterey, CA, 2002, Paper AIAA-2002-4678.

[14] A. Moitra, R. Szczerba, V. Didomizio, L. Hoebel, R. Mattheyses, and B. Yamrom, "A novel approach for the coordination of multi-vehicle teams," in *Proc. AIAA Guidance, Navigation, and Control Conf.*, Monterey, CA, 2001, pp. 608–618.

[15] K. Passino, *An Introduction to Research Challenges in Cooperative Control for Uninhabited Autonomous Vehicles* (preprint).

[16] C. Schumacher, P. Chandler, and S. Rasmussen, "Task allocation for wide area search munitions via network flow optimization," in *Proc. AIAA Guidance, Navigation, and Control Conf.*, Monterey, CA, 2001, pp. 619–626.

[17] P. Vincent and I. Rubin, "A framework and analysis for cooperative search using UAV swarms," in *Proc. ACM Symp. Applied Computing*, Nicosia, Cyprus, 2004, pp. 79–86.

[18] J.-C. Latombe, *Robot Motion Planning*, ser. The Kluwer International Series in Engineering and Computer Science, vol. 124. Boston, MA: Kluwer, Aug. 1991.

[19] C. Choo, J. Smith, and N. Nasrabadi, "An efficient terrain acquisition algorithm for a mobile robot," in *Proc. IEEE Int. Conf. Robotics and Automation*, Sacramento, CA, Apr. 1991, pp. 306–311.

[20] V. Lumelsky, S. Mukhopadhyay, and K. Sun, "Dynamic path planning in sensor-based terrain acquisition," *IEEE Trans. Robot. Autom.*, vol. 6, no. 4, pp. 462–472, Aug. 1990.

[21] A. Sankaranarayanan and I. Masuda, "Sensor based terrain acquisition: A new, hierarchical algorithm and a basic theory," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Raleigh, NC, Jul. 1992, pp. 1515–1523.

[22] J. Svennebring and S. Koenig, "Building terrain-covering ant robots: A feasibility study," *Auton. Robots*, vol. 16, no. 3, pp. 313–332, May 2004.

[23] S. Wong and B. MacDonald, "A topological coverage algorithm for mobile robots," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Las Vegas, NV, Oct. 2003, pp. 1685–1690.

[24] S. Yang and C. Luo, "A neural network approach to complete coverage path planning," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 1, pp. 718–724, Feb. 2004.

[25] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *IEEE Trans. Syst., Man, Cybern.*, vol. 19, no. 5, pp. 1179–1187, Sep./Oct. 1989.

[26] M. Gill and A. Zomaya, *Obstacle Avoidance in Multi-Robot Systems: Experiments in Parallel Genetic Algorithms*, ser. Robotics and Intelligent Systems., vol. 20. Singapore: World Scientific, 1998.

[27] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proc. Int. Conf. Robotics and Automation*, St. Louis, MO, Mar. 1985, pp. 500–505.

[28] C. DeBolt, C. O'Donnell, C. Freed, and T. Nguyen, "The BUGS (Basic UXO Gathering Systems) project for UXO clearance & mine countermeasures," in *Proc. IEEE Int. Conf. Robotics and Automation*, Albuquerque, NM, Apr. 1997, pp. 329–332.

[29] RoboMower, [Online]. Available: http://www.friendlyrobotics.com

[30] Y.-J. Oh and Y. Watanabe, "Development of small robot for home floor cleaning," in *Proc. 41st SICE Annu. Conf.*, Osaka, Japan, Aug. 2002, pp. 3222–3223. no. 7803950.

[31] S. Hert, S. Tiwari, and V. Lumelsky, "A terrain-covering algorithm for an AUV," *Auton. Robots*, vol. 3, no. 2/3, pp. 91–119, Jun. 1996.

[32] S. Spires and S. Goldsmith, "Exhaustive geographic search with mobile robots along space-filling curves," in *Proc. 1st Int. Workshop Collective Robotics*, Paris, France, 1998, pp. 1–12.

[33] I. Wagner, M. Lindenbaum, and A. Bruckstein, "Distributed covering by ant-robots using evaporating traces," *IEEE Trans. Robot. Autom.*, vol. 15, no. 5, pp. 918–933, Oct. 1999.

[34] S. Benkoski, M. Monticino, and J. Weisinger, "A survey of the search theory literature," *Nav. Res. Logist.*, vol. 38, no. 4, pp. 469–494, Aug. 1991.

[35] M. Polycarpou, Y. Yang, and K. Passino, "A cooperative search framework for distributed agents," in *Proc. IEEE Int. Symp. Intelligent Control*, Mexico City, Mexico, 2001, pp. 1–6.

[36] Y. Yang, A. Minai, and M. Polycarpou, "Analysis of opportunistic method for cooperative search by mobile agents," in *Proc. 41st IEEE Conf. Decision and Control*, Las Vegas, NV, Dec. 2002, pp. 576–577.

[37] ——, "Decentralized opportunistic learning in UAV's performing cooperative search," presented at the *Proc. AIAA Guidance, Navigation, and Control Conf.*, Monterey, CA, 2002, Paper AIAA-2002-4590.

[38] Y. Yang, M. Polycarpou, and A. Minai, "Opportunistically cooperative neural learning in mobile agents," in *Proc. Int. Joint Conf. Neural Networks*, Honolulu, HI, May 2002, pp. 2638–2643.

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

[40] M. Flint, M. Polycarpou, and E. Fernandez-Gaucherand, "Cooperative control of multiple autonomous UAV's search for targets," in *Proc. 41st IEEE Conf. Decision and Control*, Las Vegas, NV, Dec. 2002, pp. 2823–2828.

[41] M. Baum and K. Passino, "A search-theoretic approach to cooperative control for uninhabited air vehicles," presented at the *Proc. AIAA Conf. Guidance, Navigation, and Control*, Monterey, CA, 2002, Paper AIAA-2002-4589.

[42] C. Zhang and R. Ordonez, "Decentralized adaptive coordination and control of uninhabited autonomous vehicles via surrogate optimization," in *Proc. Amer. Control Conf.*, Denver, CO, Jun. 2003, pp. 2205–2210.

[43] J. Grefenstette. (1997). "The user's guide to samuel-97: An evolutionary learning system." Technical Report. [Online]. Available: citeseer.ist.psu.edu/176572.html

[44] J. Marin, R. Radtke, D. Innis, D. Barr, and A. Schultz, "Using a genetic algorithm to develop rules to guide unmanned aerial vehicles," in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, Tokyo, Japan, Oct. 1999, pp. 1055–1060.

[45] J. Hespanha, J. K. Hyoun, and S. Sastry, "Multiple-agent probabilistic pursuit–evasion games," in *Proc. 38th IEEE Conf. Decision and Control*, Phoenix, AZ, Dec. 1999, pp. 2432–2437.

[46] J. Hespanha, M. Prandini, and S. Sastry, "Probabilistic pursuit–evasion games: A one-step Nash approach," in *Proc. 39th IEEE Conf. Decision and Control*, Sydney, Australia, Dec. 2000, pp. 2272–2277.

[47] O. Laporte, "The vehicle routing problem: An overview of exact and approximate algorithms," *Eur. J. Oper. Res.*, vol. 59, no. 3, pp. 345–358, 1992.

[48] H. Ozdemir and C. Mohan, "Evolving schedule graphs for the vehicle routing problem with time windows," in *Proc. Congr. Evolutionary Computation*, La Jolla, CA, Jul. 2000, pp. 888–895.

[49] T. Takeno, Y. Tsujimura, and G. Yamazaki, "A single-phase method based on evolution calculation for vehicle routing problem," in *Proc. 4th Int. Conf. Computational Intelligence and Multimedia Applications*, Yokosuka, Japan, Oct./Nov. 2001, pp. 103–107.

[50] K. Uchimura, H. Sakaguchi, and T. Nakashima, "Genetic algorithms for vehicle routing problem in delivery system," in *Proc. Vehicle Navigation and Information Systems Conf.*, Yokohama, Japan, Aug./Sep. 1994, pp. 287–290.

[51] K. Tan, L. Lee, Q. Zhu, and K. Ou, "Heuristic methods for vehicle routing problem with time windows," *Artif. Intell. Eng.*, vol. 15, no. 3, pp. 281–295, Jul. 2001.

[52] A. Richards, J. Bellingham, M. Tillerson, and J. How, "Co-ordination and control of multiple UAVs," presented at the AIAA Conf. Guidance, Navigation, and Control, Monterey, CA, Aug. 2002, Paper AIAA-2002-4288.

[53] K. P. O'Rourke, W. B. Carlton, T. G. Bailey, and R. R. Hill, "Dynamic routing of unmanned aerial vehicles using reactive tabu search," *Military Oper. Res.*, vol. 6, no. 1, pp. 5–30, 2001.

[54] J. Ryan, T. Bailey, J. Moore, and W. Carlton, "Reactive tabu search in unmanned aerial reconnaissance simulations," in *Proc. 30th Conf. Winter Simulation*, Washington, DC, 1998, pp. 873–880.

[55] R. Murphy, "An approximate algorithm for a weapon target assignment stochastic program," in *Approximation and Complexity in Numerical Optimization: Continuous and Discrete Problems*, P. Pardalos, Ed. Boston, MA: Kluwer, 2000, pp. 406–421.

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

[57] A. Gil and K. Passino, "Stability analysis of network-based cooperative resource allocation strategies," in *Proc. 42nd IEEE Conf. Decision and Control*, Maui, HI, Dec. 2003, pp. 1206–1211.

[58] R. Beard, T. McLain, M. Goodrich, and E. Anderson, "Coordinated target assignment and intercept for unmanned air vehicles," *IEEE Trans. Robot. Autom.*, vol. 18, no. 6, pp. 911–922, Dec. 2002.

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

[60] T. McLain and R. Beard, "Trajectory planning for coordinated rendezvous of unmanned air vehicles," presented at the AIAA Guidance, Navigation, and Control Conf., Denver, CO, 2000, Paper AIAA-2000-4369.

[61] T. McLain, P. Chandler, S. Rasmussen, and M. Pachter, "Cooperative control of UAV rendezvous," in *Proc. Amer. Control Conf.*, Arlington, VA, Jun. 2001, pp. 2309–2314.

[62] C. Schumacher, P. Chandler, M. Pachter, and L. Pachter, "UAV task assignment with timing constraints via mixed-integer linear programming," in *Proc. AIAA 3rd "Unmanned Unlimited" Technical Conf., Workshop and Exhibit*, Chicago, IL, Sep. 2004, pp. 238–252.

[63] *Low Cost Autonomous Attack System (LOCAAS)*. [Online]. Available: http://www.globalsecurity.org/military/systems/munitions/locaas.htm

[64] P. Chandler and M. Pachter, "Hierarchical control for autonomous teams," in *Proc. AIAA Guidance, Navigation, and Control Conf.*, Monterey, CA, 2001, pp. 632–642.

[65] P. Chandler, M. Pachter, D. Swaroop, J. Fowler, J. Howlett, S. Rasmussen, C. Schumacher, and K. Nygard, "Complexity in UAV cooperative control," in *Proc. Amer. Control Conf.*, Anchorage, AK, 2002, pp. 1831–1836.

[66] A. Pongpunwattana, R. Rysdyk, and J. Vagners, "Market-based coevolution planning for multiple autonomous vehicles," presented at the AIAA Conf. Guidance, Navigation, and Control, Austin, TX, May 2003, Paper AIAA-2003-6524.

[67] W. Agassounon and A. Martinoli, "Efficiency and robustness of threshold-based distributed allocation algorithms in multi-agent systems," in *Proc. 1st Int. Joint Conf. Autonomous Agents and Multi-Agent Systems*, Bologna, Italy, Jul. 2002, pp. 1090–1097.

[68] B. Gerkey and M. Mataric, "Multi-robot task allocation: Analyzing the complexity and optimality of key architectures," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, Taipei, Taiwan, R.O.C., Sep. 2003, pp. 3862–3868.
[69] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *IEEE Computer*, vol. 22, no. 6, pp. 46–57, Jun. 1989.
[70] J. Bellingham, M. Tillerson, M. Alighanbari, and J. How, "Cooperative path planning for multiple UAVs in dynamic and uncertain environments," in *Proc. 41st IEEE Conf. Decision and Control*, Las Vegas, NV, Dec. 2002, pp. 2816–2822.
[71] J. Finke, K. Passino, and A. Sparks, "Cooperative control via task load balancing for networked uninhabited autonomous vehicles," in *Proc. 42nd IEEE Conf. Decision and Control*, Maui, HI, Dec. 2003, pp. 31–36.
[72] H. Durrant-Whyte, S. Majumder, M. de Battista, and S. Scheding, "A Bayesian algorithm for simultaneous localization and map building," in *Robotics Research: 10th Int. Symp.,* R. Jarvis, A. Zelinsky, Eds., Lorne, Australia, 2001, pp. 3118–3123.
[73] L. Klein, *Sensor and Data Fusion: A Tool for Information Assessment and Decision Making*. Bellingham, WA: SPIE, 2004.
[74] J. Manyika and H. Durrant-Whyte, *Data Fusion and Sensor Management: A Decentralized Information-Theoretic Approach*, ser. Ellis Horwood Series in Electrical and Electronic Engineering. Upper Saddle River, NJ: Prentice-Hall, 1994.
[75] F. Zhao and L. Guibas, *Wireless Sensor Networks: An Information Processing Approach*, ser. Morgan Kaufmann series in networking. San Mateo, CA: Morgan Kaufmann, Jul. 2004.
[76] H. Moravec, "Sensor fusion in certainty grids for mobile robots," *Artif. Intell. Mag.*, vol. 9, no. 2, pp. 61–74, Jul./Aug. 1988.
[77] T. Cormen, C. Leiserson, and R. Rivest, *Introduction to Algorithms*, 2nd ed. Cambridge, MA: MIT Press, Sep. 2001.
[78] K. Nygard, P. Chandler, and M. Pachter, "Dynamic network flow optimization models for air vehicle resource allocation," in *Proc. Amer. Control Conf.*, Arlington, VA, Jun. 2001, pp. 1853–1858.
[79] *MultiUAV Simulation (Version 1.1)*, AFRL/VACA, Dayton, OH, Oct. 2001.
[80] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*, ser. Adaptive Computation and Machine Learning. Cambridge, MA: MIT Press, Mar. 1998.

Yan Jin (S'03) received the B.S. degree in electrical engineering from the University of Science and Technology of China, Hefei, China, in 2001. She is currently working toward the Ph.D. degree in electrical engineering at the Department of Electrical & Computer Engineering and Computer Science, University of Cincinnati, Cincinnati, OH.

Her main research interest includes decentralized cooperative control in distributed mobile multiagent systems and optimization algorithms.

Yan Liao received the B.S. and M.S. degrees in electrical engineering from the Northwestern Polytechnical University, Xi'an, Shanxi, China, in 1999 and 2002, respectively. She is currently working toward the Ph.D. degree in electrical engineering at the Department of Electrical & Computer Engineering and Computer Science, University of Cincinnati, Cincinnati, OH.

Her research interests include multiagent systems and unmanned aerial vehicle (UAV) communication and cooperation.

Ali A. Minai (S'86–M'91) received the Ph.D. degree in electrical engineering from the University of Virginia, Charlottesville, in 1991.

After his postdoctoral research in computational neuroscience at the University of Virginia, he joined the Department of Electrical & Computer Engineering and Computer Science at the University of Cincinnati, Cincinnati, OH, in 1993, where he is now an Associate Professor. His current research focuses on self-organizing complex systems, including neural networks, sensor networks, swarms, and biological systems. He is a coeditor of a forthcoming book on engineering complex systems.

Dr. Minai is a member of Eta Kappa Nu, Tau Beta Pi, and Sigma Xi. He is an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS.

Marios M. Polycarpou (S'87–M'93–SM'98–F'06) received the B.A. (*cum laude*) degree in computer science and the B.Sc. (*cum laude*) degree in electrical engineering from Rice University, Houston, TX, in 1987 and the M.S. and Ph.D. degrees in electrical engineering from the University of Southern California, Los Angeles, in 1989 and 1992, respectively.

In 1992, he joined the University of Cincinnati, Cincinnati, OH, where he reached the rank of Professor of electrical & computer engineering and computer science. In 2001, he joined the newly established Department of Electrical and Computer Engineering at the University of Cyprus, Nicosia, where he is a Professor and Interim Department Head. He teaches and conducts research in the areas of systems and control, computational intelligence, neural networks, fault diagnosis, and cooperative control.

Dr. Polycarpou is currently the Editor-in-Chief of the IEEE TRANSACTIONS ON NEURAL NETWORKS. He is an Associate Editor of the *International Journal of Applied Mathematics and Computer Science* and a past Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS (1998–2003) and of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL (1999–2002). He is a Fellow of IEEE and served as Chair of the Technical Committee on Intelligent Control, IEEE Control Systems Society (2003–2005), and as Vice President for Conferences of the IEEE Computational Intelligence Society (2002–2003).