Multi-UAV Cooperative Search Using an Opportunistic Learning Method

Yanli Yang[1]
Large Power Systems Division,
Caterpillar Inc.,
Peoria, IL 61656-1875
e-mail: Yangyanl@gmail.com

Marios M. Polycarpou
Department of Electrical and Computer Engineering,
University of Cyprus,
Nicosia 1678, Cyprus

Ali A. Minai
Department of Electrical and Computer Engineering and Computer Science,
University of Cincinnati,
Cincinnati, OH 45221-0030

The control of networked multivehicle systems designed to perform complex coordinated tasks is currently an important and challenging field of research. This paper addresses a cooperative search problem where a team of uninhabited aerial vehicles (UAVs) seeks to find targets of interest in an uncertain environment. We present a practical framework for online planning and control of a group of UAVs for cooperative search based on two interdependent tasks: (i) incrementally updating "cognitive maps" used as the representation of the environment through new sensor readings; (ii) continuously planning the path for each vehicle based on the information obtained through the search. We formulate the cooperative search problem and develop a decentralized strategy based on an opportunistic cooperative learning method, where the emergent coordination among vehicles is enabled by letting each vehicle consider other vehicles' actions in its path planning procedure. By using the developed strategy, physically feasible paths for the vehicles to follow are generated, where constraints on aerial vehicles, including physical maneuverabilities, are considered and the dynamic nature of the environment is taken into account. We also present some mathematical analysis of the developed search strategy. Our analysis shows that this strategy guarantees a complete search of the environment and is robust to a partial loss of UAVs. A lower bound on the search time for any strategy and a relaxed upper bound for the proposed strategy are given. Simulation results are used to illustrate the effectiveness of the proposed strategy. [DOI: 10.1115/1.2764515]

1 Introduction

In recent years, there has been a growing interest in employing teams of uninhabited air vehicles (UAVs) for various military and civilian missions. Such team-based operations include, but are not limited to, team reconnaissance and surveillance operations, battle damage assessment, space exploration, and scientific data gathering. UAVs are particularly suitable for tasks that are considered too dangerous for human pilots. The potential advantages of UAVs over piloted missions include reducing the cost, avoiding loss of pilots, and increased maneuverability [1,2].

Cooperative control of multiple UAVs applies to settings where teams of UAVs cooperate to accomplish a common objective. It has been shown that a collaborative team of autonomous aerial and ground vehicles can provide more effective operational capabilities to accomplish hard and complex tasks than are available through independent control of each individual vehicle [3–5]. A critical problem in realizing such multivehicle systems is to develop coherent and efficient coordination and control algorithms to maneuver each vehicle so that the team as a whole can produce complex, adaptable, and flexible group behaviors. The cooperative control problems that have received extensive attention include cooperative task allocation [6,7], cooperative path planning [8,9], cooperative search [10–12], cooperative rendezvous [13], cooperative formation control [14,15], and many others.

This paper focuses on the multivehicle cooperative search problem where a team of UAVs moves in an environment of known extent seeking targets of interest and gathering information about the environment. The UAVs only have limited or nonexistent a priori information about the target distribution in the environment. Thus, they need to move through the environment using their sensors to gather information incrementally and locate targets. The decision on where to search is driven by an objective to maximize information gathering about the environment as a function of search time.

One critical aspect of search is to cover the whole search area sufficiently and efficiently in order to locate the targets. Some exhaustive coverage methods, such as the Zamboni search, ensure that UAVs pass over all points in the search area at least once. An exhaustive search is perhaps the best strategy when the environment is uniform and static (no moving targets and pop-up threats), and the UAVs have unlimited time and perfect target identification sensors. However, in many situations, time or fuel limitations may not permit covering the target environment completely. Moreover, there is always some uncertainty in the sensors' measurement, and some "pop-up" threats in the search region can only be detected when a UAV maneuvers into their proximity. In these situations, to achieve the mission with minimal human intervention, the UAVs need to possess autonomous and adaptive path planning capabilities. The main motivation for our work is to develop and analyze a systematic cooperative search strategy to enable autonomous cooperative path planning in a team of UAVs.

We are interested in the capability of UAVs working in a distributed unsupervised mode; i.e., the UAVs themselves determine where to search based on their knowledge of the environment and do not rely on external guidance. As a result, the UAVs in the team can continue to search even when one or more members of the team are lost before task completion, making this a robust strategy. While vehicles can certainly search the environment without cooperation, the search can be made much more efficient by using cooperation to minimize duplicated effort, where some vehicles may follow similar search paths and waste search effort. Therefore, the key problem for multivehicle cooperative search is for each individual vehicle to choose sufficiently different search paths.

We have previously developed a general framework for the multi-UAV cooperative search problem for directing a group of

[1]Corresponding author.

Contributed by the Dynamic Systems, Measurement, and Control Division of ASME for publication in the JOURNAL OF DYNAMIC SYSTEMS, MEASUREMENT, AND CONTROL. Manuscript received April 16, 2006; final manuscript received January 10, 2007. Review conducted by Prabhakar R. Pagilla.

UAVs to cooperatively search a dynamic and uncertain environment, and have reported a recursive approach to cooperative search using multiobjective cost function and q-step path planning algorithms [16,17]. In this paper, we present a more formal formulation of the multivehicle cooperative search problem using a discretized cellular space with air vehicles moving synchronously at a constant speed. We present a prediction-based cooperative search method, where the main idea is to let each UAV learn to predict the states of other vehicles in its neighborhood and utilize these predictions in its path planning process such that the overall information about the environment is increased as rapidly as possible. The prediction process is done using feed-forward neural networks trained by a reinforcement learning (RL) process. Alternative RL methods have been used in robotics and multiagent systems [18]. We consider the issue of centralized versus decentralized intelligence and propose an adaptive prediction process, where the UAVs share their predictors opportunistically to increase the overall performance of the team. By using the developed strategy, physically feasible paths for the vehicles to follow are generated, where constraints on aerial vehicles, including physical maneuverabilities, are considered and the dynamic nature of the environment is taken into account. We also present some mathematical analysis of the developed search strategy. Our analysis shows that this strategy guarantees a complete search of the environment and is robust to a partial loss of UAVs. A lower bound on the search time for any strategy and a relaxed upper bound for the proposed strategy are given. The effectiveness of the method is demonstrated by various simulation results.

The remainder of the paper is structured as follows. Section 2 reviews related research work. Section 3 describes the general cooperative search framework, which is the basis of this research work. Section 4 introduces the formulation of a decentralized control model for a multivehicle cooperative search problem. Based on the developed formulation, an intelligent prediction-based path planning method is discussed and an opportunistic cooperative learning (OCL) method is proposed in Sec. 5. Section 6 presents some mathematical analysis of the proposed search strategy, Sec. 7 presents the developed opportunistic learning algorithms, and Sec. 8 reports and discusses the simulation results. Section 9 concludes the paper with some final observations.

2 Related Research Work

Search problems occur in a number of military and civilian applications, such as search-and-rescue operations in the open sea or sparsely populated areas, search missions for previously spotted enemy targets, seek-destroy missions for land mines, and search for mineral deposits. A number of approaches have been proposed for addressing such search problems.

Search theory grew out of the development of theory and practice of search-and-rescue operations in World War II (WWII) [19,20]. Search theory deals mainly with the problem of distribution of search effort over an environment consisting of cells in a way that maximizes the probability of finding an object of interest. Typically, it is assumed that some prior knowledge about the target distribution is available, and so is the "payoff" function that relates the time spent on searching to the probability of actually finding the target, given that the target is indeed in a specific search region. Search theory is one of the most well established areas in operation research, and a great deal of progress has been made in this area. The solutions for most of the stationary target problems have been derived [19,20]. For the moving target problem, the emphasis in search theory has shifted from mathematical and analytical solutions to algorithmic solutions. A number of heuristic approaches that result in "approximately optimal" solutions have been proposed (see Refs. [21–23]). Detailed reviews of the current status of search theory can be found in Refs. [24,25]. So far, search theory has paid little attention to the problem of having a team of cooperating searchers, and the solutions obtained from search theory are mainly of theoretical rather than of practical value. For example, search theory usually requires that the search effort be infinitely divisible between cells, which makes it difficult to generate physically feasible flight trajectories for air vehicles.

The problem of cooperative search is essentially one of planning efficient search paths—paths that yield a maximum of new information. Thus, the vehicles need to possess autonomous cooperative path planning capabilities in order to perform cooperative search tasks. The cooperative path planning problem has received a great deal of research attention in the cooperative control area. Related UAV cooperative control problems employing cooperative path planning includes cooperative classification [8], cooperative attack [16,26], cooperative rendezvous [13], and cooperative task assignment [6,7,9,27–29]. The cooperative path planning considered in these problems involves timing or sequencing of UAVs for arrival at targets or other specified locations. Thus, the path planning problem is reduced to that of finding a flyable path from a UAV's initial position to its destination. However, these destination-oriented (or point-to-point) path planning methods cannot be utilized directly for the path planning involved in the search task since the vehicles do not have an assigned destination and they may have to treat all uncertain areas as possible destinations in order to gather information.

The search path planning problem shares a lot of commonalities with the mobile robot coverage path planning problem, where one or more mobile robots are required to explicitly pass over all points in an unexplored region with stationary obstacles to accomplish some tasks, such as floor cleaning, lawn mowing, mine hunting, and harvesting. Many exhaustive coverage path planning methods, e.g., Zamboni search, have been developed [30–35] and Ref. [36] provides a good survey of major results in this area. Though these coverage algorithms provide a good source of intuition for UAV search path generation, most of them do not consider the constraints on aerial vehicles and the dynamic nature of the search environment. For example, in many situations, the time or power limitations may not permit the UAVs to cover the whole environment completely. The physical maneuverability may constrain the minimum turning radius of the vehicle and prohibit the vehicles from making a severe G turn, while most of the exhaustive coverage algorithms require agents to move in any direction as desired. Moreover, as the mission is executed, the information about the environment and the environment itself may change unpredictably due to the various uncertainties, such as imperfect sensor accuracy and pop-up threats, whereas many of the coverage algorithms are based on the assumption of an environment with stationary obstacles. These factors have limited the performance of applying coverage algorithms in solving the UAV cooperative search problem [16].

The cooperative search problem has attracted significant interest recently due to its importance in a variety of applications. Only after vehicles have done some search and have obtained necessary information about the environment is it possible to initiate other tasks, such as cooperative classification, cooperative attack, cooperative engagement. A general framework for directing a group of UAVs to cooperatively search a dynamic and uncertain environment is developed in Refs. [16,17], where the search path generation problem is separated into two parts: the online environment modeling process and the real-time path decision process. The coordination among vehicles is achieved by considering the group benefit in each vehicle's decisions. Based on this framework, a Bayesian map-building method used to probabilistically model the environment is proposed, and a cooperative learning method to achieve the team autonomy is discussed in Refs. [10,37–39]. A comparison between two evidential map-building methods is presented in Ref. [40]. Using the same framework, a dynamic programming based cooperative search path planning method is developed in Refs. [11,41–43], a k-shortest path algorithm based

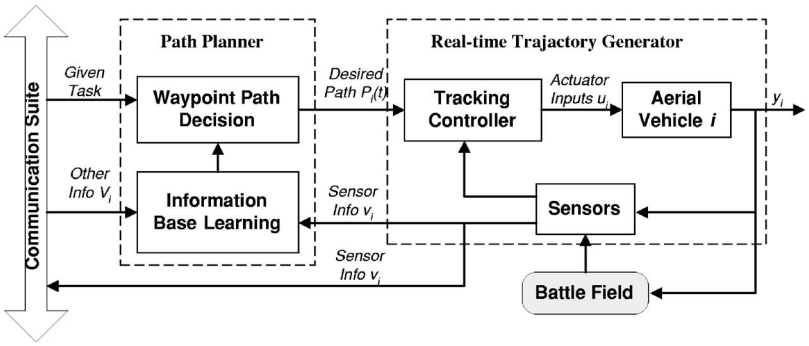

Fig. 1 A general cooperative search framework

search path generation method is studied in Refs. [44,45], and a game theoretic search scheme is proposed in Refs. [46,47]. Reference [48] presents an approach to apply optimal search theory to solve the multivehicle cooperative search problem. Reference [49] addresses the cooperative search problem using a surrogate optimization based decentralized control algorithm to yield UAV search trajectories in a coordinated manner. Some exhaustive coverage algorithms have also been applied to solve the cooperative search problem [12,50], where Ref. [12] proposes advanced formation coverage methodologies, which could be robust to the loss of UAVs during search.

3 Cooperative Search Framework

This section gives a brief introduction to a general cooperative search framework. The proposed framework for online planning and control of the UAVs to perform the cooperative search task is illustrated in Fig. 1. In this framework, rather than considering the full dynamic model when calculating guidance trajectories, each UAV uses two processes in guidance and control: the high-level *path planner* and the low-level *real-time trajectory generator*. The path planner maintains an information base about the current status of the environment and uses optimal control methods to generate a waypoint path, $P_i(t)$, for the vehicle to follow. The waypoint path refers to a series of geodetic coordinates in the environment through which the UAV must travel. In order to generate a feasibly flyable waypoint path for a vehicle, the air vehicle maneuverability constraints (such as limited turning angle) and pop-up threats must be accounted for. Given the predetermined waypoint path, the real-time trajectory generator uses a dynamics model of the air vehicle to produce the actuator input u_i so that the vehicle tracks the desired trajectory $P_i(t)$. In this paper, we largely ignore the vehicle dynamics and concentrate on the path planner design problem. Our focus is on showing how the resident information of each vehicle can be combined with information from other vehicles so that the team of vehicles can work together to reach the group objective.

The design of the outer-loop control scheme is broken down into two basic functions, as shown in the left block in Fig. 1. First, the UAV uses the sensor information received to update its information base or cognitive map, which is a representation of the environment, as well as other information, such as its location and direction, the location and direction of the other vehicles, remaining fuel. This will be referred to as the UAV's *update* function. Based on its information base, the UAV computes a desired path to follow. This is called the UAV's *decision* function. The vehicle uses a cognitive map in the form of a Cartesian grid as its environment representation, where each cell is assigned a value representing the vehicle's knowledge about the target/threat distribution in the corresponding region. As the UAVs search the environment, their cognitive maps are continuously updated by incorporating the vehicles' sensor readings through an evidential sensor fusion algorithm, where inaccuracies and uncertainties in the sensor information are taken into consideration. As the cognitive maps are built incrementally, the waypoint path decision function in each UAV utilizes its map to estimate the rewards expected from searching certain areas of the environment and to generate a feasible path to direct the vehicle's search. This is achieved through an abstract UAV dynamics model taking into consideration the vehicle's maneuverability constraints, and then defining an optimal control problem formulation, which is used to generate a maximum-reward path.

In this setting, we assume that the guidance control decisions made by each UAV are autonomous, in the sense that no UAV tells another what to do in a hierarchical type of structure, nor is there any negotiation between UAVs. Each UAV simply receives information about the environment from the other vehicles (or a subset of the remaining vehicles) and makes its own decisions, which are typically based on enhancing a global goal, not only its own goal. Therefore, the presented framework can be thought of as a *passive cooperation* (or stigmergic coordination) framework, as opposed to *active cooperation*, where the UAVs may be actively coordinating their decisions and actions. For simplicity, we assume that the vehicles are flying at different altitudes so that they do not collide when two or more vehicles fly into the same cell at the same time. Therefore, we do not explicitly address the collision avoidance problem among vehicles in this framework.

4 Cooperative Search Problem Formulation

Using the general framework discussed in Sec. 3, this section presents the theoretical formulation of the cooperative search problem, including the environment model, the vehicle dynamics model, the map-building method, and the cooperative path planning decision process.

4.1 The Environment. The *environment E* is a bounded $L_x \times L_y$ cellular area, where each position is termed a *cell*. Each cell (x,y) has an associated *uncertainty* value, $z(x,y,t) \in [0,1]$, representing the UAVs' uncertainty about the target distribution in that cell. If $z(x,y,t)=1$, then cell (x,y) is a completely unknown location for the vehicles at time t. As the cell is searched repeatedly, $z(x,y,t)$ approaches 0. A cell (x,y) is said to be *fully searched* if $z(x,y,t) \leq C$, where C is a predefined threshold corresponding to a decision that the cell does not need to be searched any more. The uncertainty value associated with each cell represents the undetected information in that location.

There is a team of N identical UAVs moving synchronously in discrete time and continually sensing the environment using their sensors. The vehicles are assumed to be equipped with reliable communication capabilities so that they can exchange sensing information among the group without any error or delay. The objective of the group of vehicles is to search the environment coop-

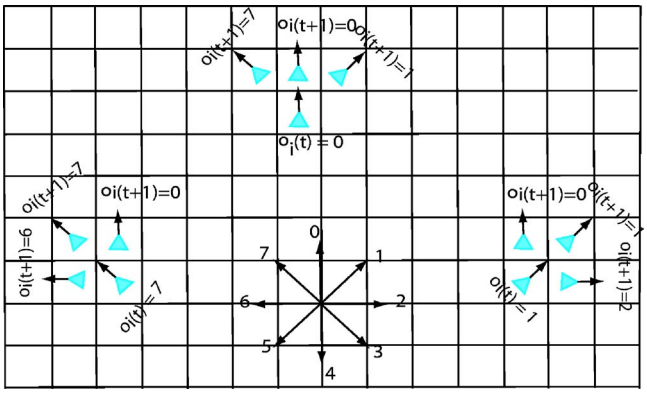

Fig. 2 Example orientation transition choices for UAVs

eratively in order to gather as much information as possible, minimizing their ignorance about the environment as a function of time. Since the more information the UAV have about one cell, the lower the cell's uncertainty value will be, the vehicles' ignorance about the whole environment can be quantitatively represented by the sum of the uncertainty values of all cells in the environment, which is defined as the total uncertainty about the environment.

4.2 Simplified Uninhabited Aerial Vehicle Dynamics Model. At each time step, a UAV can move from its current cell to one of the eight neighboring cells, subject to some maneuverability constraints. The state of UAV i at time t is denoted by $v_i(t)$, which is comprised of two components: $v_i(t)=[\lambda_i(t),o_i(t)]$. The first component $\lambda_i(t)=[x_i(t),y_i(t)]\in\{1,2,\ldots,L_x\}\times\{1,2,\ldots,L_y\}$ is the vehicle's position in the environment at time t. The second element $o_i(t)\in\{0,1,\ldots,8\}$ is the vehicle's orientation defined as {0 (north), 1 (northeast), 2 (east), 3 (southeast), 4 (south), 5 (southwest), 6 (west), or 7 (northwest)}.

The UAV's dynamics is subject to physical curvature radius constraints, reflected in the fact that it can only change its orientation by at most one step, that is, $o_i(t+1)\in[o_i(t)-1,o_i(t),o_i(t)+1]\bmod 8$. In this way, each UAV has three possible positions for the next time step, i.e., turn 45 deg left, turn 45 deg right, or go straight, designated by l (left), r (right), and f (forward), respectively. The control decision for UAV i is its path selection at each time step t, denoted by $u_i(t)\in\{l,f,r\}$. Figure 2 shows this graphically for various orientations.

In our model, UAVs use a q-steps-ahead path planning method (see Ref. [16]); that is, each UAV plans its path q steps ahead of its current location, adding a new move at each time step. The main advantage of a plan-ahead algorithm is that it creates a buffer for path planning. From a practical perspective, this can be quite useful for air vehicles that require (at least) some trajectory planning. For simplicity, we use $q=1$ in this study (and the extension to the case of $q>1$ is straightforward). Thus, UAV i at time step t executes an *action* comprising the following three steps.

1. It makes decision $u_i(t+1)$ to choose a new orientation $o_i(t+2)$ for time step $t+2$.
2. It then determines its step $t+2$ position $\lambda_i(t+2)$ as the neighbor of $\lambda_i(t+1)$ in the direction corresponding to orientation $o_i(t+2)$.
3. Finally, it executes its decision $u_i(t)$ for the current time step t and updates its state $v_i(t+1)=[\lambda_i(t+1),o_i(t+1)]$ by moving to cell $\lambda_i(t+1)$ with orientation $o_i(t+1)$.

In summary, a vehicle's transition function can be expressed as

$$v_i(t+1)=f_v[v_i(t),u_i(t)] \tag{1}$$

Using this formulation, the cells that a UAV chooses to search in sequence constitute a waypoint path that can be used as a reference path for the vehicle to determine its real-time flight trajectory and to determine in which region its search effort should be expended.

4.3 Uninhabited Aerial Vehicle Information Base. In our framework, each vehicle has an *information base* to represent and store its knowledge, which consists of an environment model (in the form of a "cognitive map") and a *vehicle state vector*. This information is used to guide the search to certain areas of the environment and can be consistently updated using sensor information about the environment.

In this section, we give a detailed description of the information base and present a simplified evidential map-building method to incrementally update the cognitive maps. The basic idea of the map-building method is to represent sensor observations probabilistically and then use an evidential method to fuse sensor information.

As discussed earlier, each cell (x,y) has an associated uncertainty value, $z(x,y,t)\in[0,1]$, representing the degree of ignorance about the contents in that location. Each UAV i uses a cognitive map to store its knowledge about the uncertainties in the environment and continuously updates it using new sensor readings from its own or from other vehicles by communication. This kind of cognitive map is defined as an *uncertainty map* and is denoted as $\mathcal{Z}^i(t)$. Each cell in the uncertainty map is initialized with a value in $[C,1]$ to reflect the vehicle's a priori knowledge about that location. A cell with $z^i(x,y,0)=1$ is completely unknown to UAV i and needs to be searched. A cell with $z^i(x,y,0)=C$, where C is a constant between 0 and 1, is a location with no interest for search (for example, a location in a lake would be initialized as C if the targets are all land based). In general, the uncertainty map carried by different vehicles could be different because of the different search history of each vehicle and the information loss caused by the communication failure or delays. In this study, we assume for simplicity that the communication among the group of UAVs is reliable and the sensor information from any UAV is available to the whole group immediately, so all UAVs share the same uncertainty map, which is denoted as $\mathcal{Z}(t)$. This is a restrictive assumption, which will be relaxed in future reports.

We consider a case of imperfect sensors in this study; i.e., each sensor scan does not by itself provide a 100% certainty about the state of the corresponding location. Here, we define an *uncertainty reduction rate*, denoted as $\eta_i\in(0,1]$, to model the uncertainties and inaccuracies in UAV i's sensor. Mathematically, η_i quantifies the belief of a sensor scan from vehicle i committed to reducing the uncertainty in that cell. Since the vehicles are identical in this study, we use η to denote the uncertainty reduction rate for all the vehicles. Based on the defined sensor model and using Dempster's rule of combination [51], a rule for updating the uncertainty map can be described as follows (see Appendix for a detailed derivation): a visit by any vehicle to cell (x,y) at time t reduces the uncertainty value associated with that cell at a rate η, represented as

$$z(x,y,t+1)=\eta z(x,y,t) \tag{2}$$

It is easy to generalize the above rule to the case where m vehicles visit the cell at the same time; the cell's uncertainty value is updated as

$$z(x,y,t+1)=\eta^m z(x,y,t) \tag{3}$$

According to Eqs. (2) and (3), it is easy to see that the first scan of a cell results in the maximum reduction in uncertainty, and further scans have diminishing returns. For example, if $\eta=0.5$, the uncertainty value of a cell (x,y) with $z(x,y,0)=1$ changes as $1\mapsto 0.5\mapsto 0.25\mapsto 0.125\mapsto 0.0625$ if it is sequentially visited four times by (possibly different) vehicles. Therefore, this update rule

is a simple way to track the number of useful "looks" each cell has had and captures the nature of diminishing returns with each look. This property is similar to that of the detection function used in search theory (see, for example, Refs. [19,20]), where the detection function represents the probability that a search in a given cell for a specified duration of time will detect the target, provided that the target is present in that cell and each incremental time unit spent in searching a cell produces a decreasing return on the probability of detection.

The vehicle state vector is used to store a UAV's knowledge about the group of UAVs' states and decisions. We define $\mathbf{v}^i(t)=[v_1^i(t),v_2^i(t),\ldots,v_N^i(t)]$ to represent vehicle i's knowledge about all N vehicles' states at time t, where $v_j^i(t)=[\lambda_j^i(t),o_j^i(t)]$, $j=[1,2,\ldots,N]$ denotes the vehicle j's state at time t according to vehicle i's knowledge. $\mathbf{u}^i(t)=[u_1^i(t),u_2^i(t),\ldots,u_N^i(t)]$ is defined to represent vehicle i's knowledge of other N vehicles' decisions made for time t, where $u_j^i(t)\in\{l,f,r\}$ denotes vehicle j's decision at time t stored in vehicle i's information base. As before, $v_j^i(t)$, $u_j^i(t)$, in principle, might be different from $v_j^k(t)$, $u_j^k(t)$ for $i\neq k$ because of information loss. Due to the perfect communication assumption in this study, the vehicle state vectors are actually the same for each vehicle, which are denoted as $\mathbf{v}(t)=[v_1(t),v_2(t),\ldots,v_N(t)]$ and $\mathbf{u}(t)=[u_1(t),u_2(t),\ldots,u_N(t)]$.

4.4 Waypoint Path Decision Process. Given the current information available via the information base, each UAV needs to evaluate the cost function associated with each path and select an optimal path to follow in order to accomplish the search task most efficiently. The decentralized waypoint path decision problem for each vehicle can be formulated into an optimal control problem as follows.

A UAV's knowledge about the environment stored in its information base (composed of its uncertainty map and its vehicle state vector) can be defined as its *control state* for its path selection decision process. As discussed before, the perfect communication assumption used in this study means that all UAVs share the same control state, and this shared control state can be represented as

$$\mathbf{x}(t)=[\mathcal{Z}(t),\mathbf{v}(t),\mathbf{u}(t)]$$

where $\mathcal{Z}(t)$, $\mathbf{v}(t)$, and $\mathbf{u}(t)$ are defined in Sec. 4.3. As we know, UAV i's decision at time t is to select its path at time $t+1$, denoted as $u_i(t+1)\in\{l,f,r\}$, where l means turn left, r means turn right, and f means go straight. We define a decision vector $\mathbf{u}(t+1)=[u_1(t+1),u_2(t+1),\ldots,u_N(t+1)]$ to represent all the vehicles' decisions at time $t+1$. It can be seen that $\mathbf{u}(t+1)$ is a function of the control state at time t, given by

$$\mathbf{u}(t+1)=h[\mathbf{x}(t)] \tag{4}$$

where the form of the decision function $h(\cdot)$ is determined by the path selection strategies used by the group of vehicles. After UAVs make their decisions for time $t+1$, they execute their decisions $\mathbf{u}(t)$, scan the cells that they are moving into, and update their uncertainty maps. The vehicles' actions, in consequence, determine a new control state at time $t+1$, $\mathbf{x}(t+1)$. The transition function is denoted as

$$\mathbf{x}(t+1)=f_s[\mathbf{x}(t),\mathbf{u}(t)] \tag{5}$$

where each function $f_s(\cdot)$ is determined by the specific moving rules and uncertainty update rules taken by the UAVs. Equations (4) and (5) define the dynamics of the system so that the decisions by the team of UAVs cause the transitions of the control state, which, in turn, affect the decisions of vehicles.

The objective of the search mission is to minimize the uncertainty over the whole environment for a finite time horizon of T_f time units. This can be achieved by the UAVs cooperatively planning their paths such that the following payoff function is minimized:

$$G[\mathbf{x}(T_f)]:=\sum_{(x,y)\in E} z(x,y,T_f) \tag{6}$$

where Eq. (6) represents the total uncertainty at the end of the search task (time T_f).

Dynamic programming [52] is one possible approach for this optimal path selection problem as a function of the control state and the remaining time to complete the task. Because the dimension of the state space is typically quite large for realistic search problems, solving the optimal control problem by dynamic programming methods is computationally prohibitive. This calls for approximate methods, and in this study, we develop an intelligent learning method using estimated reward functions, where the cooperation among UAVs is achieved by letting each UAV learn to predict and consider other UAVs' state for its decision.

5 Cooperative Search Path Planning Strategy

In this section, we describe a path selection decision function $h[\mathbf{x}(t)]$ for UAVs to perform the cooperative search task. The decision function is based on the expected reward associated with each possible path for the UAV to follow.

According to the vehicle dynamics described in Sec. 4.2, at time t, UAV i has a position $[x_i(t),y_i(t)]$ and has decided its control action $u_i(t)$, indicating which direction it will follow at the next time step (which was decided at time $t-1$). The main task for its path selection decision is to choose a direction to follow at time $t+1$, denoted by $u_i(t+1)\in\{l,f,r\}$, which, in turn, will decide its position at time $t+2$, $[x_i(t+2),y_i(t+2)]$. The three different directions (l, turn left; r, turn right; and f, go straight) will direct the vehicle to three different target cells neighboring its position at $t+1$. Based on ith UAV's current position $[x_i(t),y_i(t)]$, its decision $u_i(t)$, and its moving rule defined in Sec. 4.2, its position at time $t+1$ $[x_i(t+1),y_i(t+1)]$ is known. Consequently, its three target cells for $t+2$ can be determined, which are denoted as $[x_i^l(t+2),y_i^l(t+2)]$, $[x_i^f(t+2),y_i^f(t+2)]$, and $[x_i^r(t+2),y_i^r(t+2)]$. For simplicity, we use k_i to denote one of the ith UAV's target cells with position $[x_i^k(t+2),y_i^k(t+2)]$, where $(k\in\{l,f,r\})$. UAV i uses the following path decision strategy to determine which cell to go to: It estimates (predicts) the expected reward for each target cell k_i and selects the one with the best payoff, or reward.

5.1 Reward Estimation. In this study, we consider two methods for estimating the reward for a potential target cell k_i.

Greedy estimation. In this case, the UAV i assumes that it will be able to capture the entire current reward available at the target cell k_i $(k_i=[x_i^k(t+2),y_i^k(t+2)]$, $k\in\{l,f,r\}$ when it moves into this location at time $t+2$. Thus, the estimated reward in cell k_i is

$$\hat{\rho}_{k_i}(t+2)=(1-\eta)z(k_i,t)$$

where $z(k_i,t)$ is the uncertainty value in cell k_i at time t. This simple estimate ignores two issues. First, it is possible that other vehicles may visit k_i at step $t+1$, thus reducing the reward available to i at $t+2$. Second, other vehicles may enter k_i at $t+2$, thus diluting the reward i receives. We use this simple estimate primarily to determine a reasonable reference base line for system performance.

Cooperative estimation. This estimate takes into account both of the above factors ignored by the greedy estimate. It is determined as follows. Suppose $\nu_{k_i}(t+1)$ UAVs occupy cell k_i at $t+1$ and $\nu_{k_i}(t+2)$ (including i) at step $t+2$. Then, we get

$$z(k_i,t+1) = \eta^{\nu_{k_i}(t+1)} z(k_i,t)$$

and

$$z(k_i,t+2) = \eta^{\nu_{k_i}(t+1)+\nu_{k_i}(t+2)} z(k_i,t)$$

The reward obtained by i in this case would be

$$\rho_{k_i}(t+2) = \frac{1}{\nu_{k_i}(t+2)}[z(k_i,t+1) - z(k_i,t+2)]$$
$$= \frac{\eta^{\nu_{k_i}(t+1)}[1-\eta^{\nu_{k_i}(t+2)}]}{\nu_{k_i}(t+2)} z(k_i,t) \quad (7)$$

It can be seen that the reward is determined by three factors: (1) the uncertainty value in cell k_i at time t, $z(k_i,t)$; (2) the occupancy of cell k_i at $t+1$, $\nu_{k_i}(t+1)$; and (3) the occupancy of cell k_i at $t+2$, $\nu_{k_i}(t+2)$. Clearly, $z(k_i,t)$ is available from the uncertainty map; moreover, since all UAVs decide their move for step t at time $t-1$ and they communicate its decision right away, $\nu_{k_i}(t+1)$ is, in fact, determined at step $t-1$ and is already known by each vehicle at time t. Thus, only $\nu_{k_i}(t+2)$ needs to be estimated for $k \in \{l,f,r\}$.

5.2 Cell Occupancy Estimation. As described above, the primary estimation problem solved by UAV i in order to decide its move at time $t+2$ is the occupancy, $\nu_{k_i}(t+2)$, for all the reachable cells, k_i. This is done based on six items of information:

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

$$c_{k_i}(t+1) = \frac{1}{\beta}|C_1[x_i^k(t+2), y_i^k(t+2), t+1]|$$

where $C_1(x,y,t)$ denotes a set of UAVs that can reach cell (x,y) at time $t+1$, $|.|$ denotes cardinality, $x_i^k(t+2)$ and $y_i^k(t+2)$ are the coordinates of target cell k_i, and β is a scaling constant (we use $\beta=8$).

Together, these two sets of values define a *state* used by UAV i, $s_i(t)=[\nu_{l_i}(t+1), \nu_{f_i}(t+1), \nu_{r_i}(t+1), c_{l_i}(t+1), c_{f_i}(t+1), c_{r_i}(t+1)]$. Note that all the c and ν values for $t+1$ are available to UAV i at time t because the UAVs communicate their positions and decisions to each other without any delay. We use a neural network consisting of three independent sub-networks to estimate $\nu_{l_i}(t+2)$, $\nu_{f_i}(t+2)$, and $\nu_{r_i}(t+2)$ using $s_i(t)$ as the input (see Sec. 7). The predicted values are denoted as $\hat{\nu}_{l_i}(t+2)$, $\hat{\nu}_{f_i}(t+2)$, and $\hat{\nu}_{r_i}(t+2)$. It should be noted that the state information available to each UAV is an extremely incomplete view of the whole environment state. More informative state formulations can be envisioned (e.g., uncertainty values for all neighbors of target cells); however, this will increase the complexity of the learning problem.

5.3 Path Decision Function. The predicted $\hat{\nu}_{k_i}(t+2)$ is used in Eq. (7) along with the known values of $z(k_i,t)$ and $\nu_{k_i}(t+1)$ to obtain an estimate of the two-step reward,

$$\hat{\rho}_{k_i}(t+2) = \frac{\eta^{\nu_{k_i}(t+1)}[1-\eta^{\hat{\nu}_{k_i}(t+2)}]}{\hat{\nu}_{k_i}(t+2)} z(k_i,t) \quad (8)$$

and the UAV chooses the cell that promises the greatest reward,

$$u_i(t+1) = k_i \quad (9)$$

where

$$\hat{\rho}_{k_i}(t+2) = \max[\hat{\rho}_{l_i}(t+2), \hat{\rho}_{f_i}(t+2), \hat{\rho}_{r_i}(t+2)]$$

After the UAV i executes its decision and moves into the target cell k_i at time $t+2$, it will receive the real reward given by Eq. (7).

By using this decision strategy, the UAVs can take into account other vehicles' possible actions when they make decisions on where to search, thus minimizing the potential overlap in information gain amongst them.

6 Performance Analysis of Cooperative Search Method

In this section, we present some mathematical analysis of the cooperative search strategy developed in Sec. 5. Specifically, we prove the completeness of the algorithm and give lower and upper bounds on its search time.

The problem of cooperative search is essentially equivalent to deriving search paths for UAVs in order to reduce the total uncertainty as fast as possible. The analysis of the cooperative search strategy considers several characteristics. The first obvious one is the property that by utilizing a designated strategy, the UAVs can indeed accomplish their task and fully search each unknown cell in the environment. We define this characteristic as *completeness* and use the following proposition to prove the completeness property of the proposed algorithm.

PROPOSITION 1. *The developed cooperative search strategy guarantees a complete search of an arbitrarily unknown environment E given enough time.*

Proof. We prove the above statement by contradiction. If this strategy is not to lead to a complete search, then all UAVs must get captured by periodic paths. Let P_i denote a set of cells that consist of a periodic path followed by UAV i and let $p_i(n)$ denote a cell belonging to this path, $p_i(n) \in P_i$. Then, the three neighboring cells that can be reached from $p_i(n)$ at the next time step are denoted as $[p_i^l(n), p_i^f(n), p_i^r(n)]$. According to the decision procedure described above,

$$p_i(n+1) \in [p_i^l(n), p_i^f(n), p_i^r(n)] \quad (10)$$

where $p_i(n+1)$ is the cell following $p_i(n)$ on the periodic path P_i. From Eq. (2), we know that the reward for searching a cell diminishes as the number of visits increase. Thus, after being visited enough times, the cell $p_i(n+1)$ will not be the cell with the maximum obtainable reward in $[p_i^l(n), p_i^f(n), p_i^r(n)]$; i.e., the available rewards in the other two cells will be better than the reward available in $p_i(n+1)$. This will cause UAV i to choose another cell to gain better reward and deviate from cell $p_i(n+1)$. Thus, the periodic path is broken up. This shows that no periodic path is possible for an infinite duration of time, which proves the result that a complete search of E will be accomplished in finite time. □

A second important characteristic of a cooperative search strategy is the duration of time required to fully search the environment E, defined as *search time* in this study. We show that there is a lower bound on the search time needed to fully search a given environment, independent of the strategy used for planning paths.

PROPOSITION 2. *The length of the search time t_s for a team of N UAVs to fully search environment E is bounded as follows*:

$$t_s \geq \frac{(A-I)}{N} \lceil \log_\eta C \rceil \quad (11)$$

where C is the predefined uncertainty value denoting a full search criterion, A is the total number of cells in E, and I is the number of cells with initial uncertainty value C.

Proof. Every scan made by a UAV in a cell will decrease the uncertainty value in that cell until it reaches the minimum uncertainty level possible. Let T_c denote the smallest number of times that a cell with an initial uncertainty value 1 needs to be scanned

before the cell becomes fully searched (i.e., its uncertainty value is smaller than C). Using Eq. (2), we obtain $T_c=\lceil \log_\eta C \rceil$. In all, there are $A-I$ cells in the environment E with an initial uncertainty value 1, and the other I cells are initialized with an uncertainty value C denoting that they are of no interest for search. Thus, the team of N UAVs needs to do at least $T_c(A-I)$ scans to fully search every cell in the environment E. However, as defined, one UAV can only scan one cell at each time step and the scan can only be contributing to complete the search task if the cell being scanned has not been fully searched. Thus, after time t_s, the team of N UAVs can at most have made NT_s effective scans of E. By equating Nt_s to $T_c(A-I)$, the result is implied. □

Next, we give an upper bound on the search time for our proposed cooperative search algorithm under a specific assumption. We begin with some basic definitions and present some preliminary results that will be used in the proof later.

Given any two cells $p_u=(x_u,y_u)\in E$ and $p_v=(x_v,y_v)\in E$, an *unconstrained path* from p_u to p_v is defined as a path that can be followed by a UAV with no maneuverability constraint, which means that the UAV can move to any of its eight-neighbor cells. The length of the shortest unconstrained path from p_u to p_v is denoted as $d_N(u,v)$. It is obvious that

$$d_N(u,v)=\max(|x_u-x_v|,|y_u-y_v|) \tag{12}$$

A *constrained path* between p_u and p_v is defined as a path by which a vehicle with the orientation constraint (described in Sec. 4.2) can move from p_u to p_v. The formation of a constrained path is decided by the relative positions of p_u and p_v and also by the UAV's initial orientation in cell p_u. When the UAV starts with an orientation $o\in[0,1,\ldots,8]$ at cell p_u, the length of the shortest constrained path for the vehicle to move from p_u to p_v is denoted as $d_C^o(u,v)$. Let $d_C(u,v)=\max_o d_C^o(u,v)$. Then $d_C(u,v)$ represents an upper bound on the length of the shortest path from p_u to p_v, where the UAV can start with any initial orientation at cell p_u. Examples of unconstrained path and constrained paths from p_u to p_v are shown in Fig. 3.

PROPOSITION 3.

$$d_C(u,v)\leqslant d_N(u,v)+8 \tag{13}$$

Proof. In Ref. [53], Dubins proved that, in a continuous space without any obstacle, a curvature-constrained shortest path from any start position to any final position consists of at most three segments, each of which is either a straight line or an arc of a unit-radius circle. Using this idea for a cellular space, we get a turn-straight-turn procedure (shown in the box below) to find a constrained path from p_u to p_v with a given initial orientation $o\in[0,1,\ldots,8]$.

Let $d_E^o(u,v)$ denote the length of path determined by the turn-straight-turn procedure. Similarly, define $d_E(u,v)=\max_o d_E^o(u,v)$. It can be shown that

$$d_E(u,v)\leqslant d_N(u,v)+8 \tag{14}$$

where equality holds only when $p_u=p_v$. The proof can be obtained by enumeration. Because the developed turn-straight-turn procedure cannot guarantee finding the shortest constrained paths for all initial orientations, we have

$$d_C(u,v)\leqslant d_E(u,v) \tag{15}$$

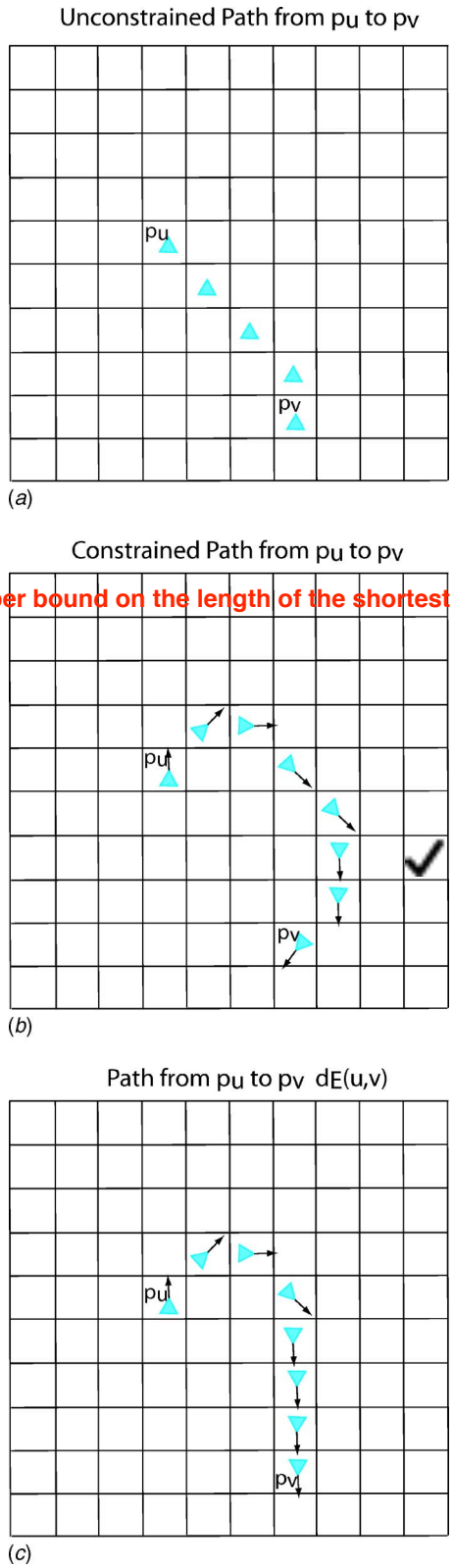

Fig. 3 Illustration of an unconstrained path, a constrained path, and a path obtained by the *turn-straight-turn* procedure from p_u to p_v

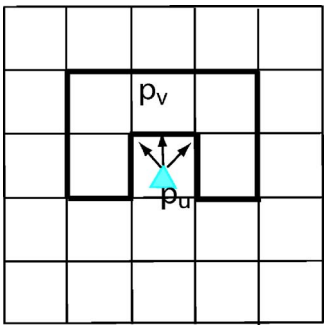

Fig. 4 Illustration of the three orientations related to p_v (as arrows) and five target cells

Based on Eqs. (14) and (15), the result is concluded. □

Turn-straight-turn procedure:

1. $t=0$;
2. $t:=t+1$;
3. compute the angle difference θ between the UAV's current orientation and the direction of a vector from the UAV's occupancy cell to p_v;
4. if p_v is one of UAV's three target cells
   go to p_v; then stop;
   else
   if $\theta<45°$
   go straight; then go to step 2;
   else
   if $t>8$
   turn left; then go to step 2;
   else
   turn right; then go to step 2;
   end
   end
   end

COROLLARY 1.

$$d_C(u,v) \leq L+8 \tag{16}$$

holds for any given $p_u, p_v \in E$, *where* $L=\max(L_x,L_y)$.

Proof. Because the environment E consists of a $L_x \times L_y$ grid, according to Eq. (10), we obtain

$$d_N(u,v) = \max(|x_u - x_v|, |y_u - y_v|) \leq \max(L_x,L_y) = L \tag{17}$$

According to Eq. (13), we get $d_C(u,v) \leq d_N(u,v)+8 \leq L+8$. □

Next, we will prove that under a specific assumption, the search time t_s needed by a group of N UAVs to fully search the environment E by following the proposed cooperative search strategy has a given upper bound.

ASSUMPTION 1. If a cell is visited m times, the number of visits, m_q with orientation $q \in 0,1,2,\ldots,7$ satisfies $|m_q - m_r| \in \{0,1\} \forall q,r \in \{0,1,2,\ldots,7\}$; i.e., all orientations happen as equally as possible, and the number of visits with each one differs by no more than 1 from the number of the visits used by others.

PROPOSITION 4. *Under Assumption 1*

$$t_s \leq \frac{A\lambda^{L+8}}{N}\lceil \log_\eta C \rceil \tag{18}$$

where A is the total number of cells in E, $\lambda=\lceil 40/3 \rceil$, *and* $L=\max(L_x,L_y)$.

Proof. The proof of this proposition is inspired by Ref. [54], but the model and assumptions are different. If cell p_u and cell p_v are neighbors in E, then p_v could be one of p_u's target cells when a UAV takes three of eight-orientations when it is in cell p_u, which are shown in Fig. 4. We call these three orientations *being related to* p_v. For these three orientations together, there are, in total, five potential target cells. According to Assumption 1, if there have been $\lambda=\lceil 40/3 \rceil$ visits to cell p_u, at least $\lfloor \lambda 3/8 \rfloor=5$ of them are with one of these three orientations related to p_v. Thus, it can be shown that cell p_v must be visited at least once every λ visits to p_u. Since after a UAV with one of the three orientations related to p_v visits p_u, one of the five target cells (shown in Fig. 4) will be visited and the reward in that cell will decrease. Thus, if after λ visits to p_u, p_v has not been visited so far, the reward of the other four target cells is less than the reward on p_v, and p_v will be visited no later than after the next visit to p_u with the orientation related to p_v. Thus, we obtain

$$S(p_v) \leq \lambda[S(p_u)+1] \tag{19}$$

where $S(p_v)$ denotes the number of visits to cell p_v.

Let us assume that a cell p_1 has not been visited; i.e., $S(p_1)=0$. Now consider the farthest cell in E from p_1, and denote it as p_l and a shortest path between them as $P=p_1,p_2,\ldots,p_l$. According to Corollary 1, we get $l \leq L+8$ since l is the length of path P. Using Eq. (19), it can be shown that

$$S(p_l) \leq \lambda + \lambda S(p_{l-1}) \leq \lambda + \lambda^2 + \lambda^2 S(p_{l-2}) \leq \cdots \leq \lambda^{L+8} + S(p_1) \tag{20}$$

Because $S(p_1)=0$, we have $S(p_l) \leq \lambda^{L+8}$. This result shows that if there is a cell in environment E that has not been searched, the largest number of visits to any other cells is bounded by λ^{L+8}. This is equivalent to saying that if one of the cells in E has been searched λ^{L+8} times, then we can say that all of the cells in E have been searched at least once. There are A cells in E, and each cell needs to be scanned at most $\lceil \log_\eta C \rceil$ times, so the maximum total number of visits needed to fully search the whole environment E is $A\lambda^{L+8}\lceil \log_\eta C \rceil$. Since we have N UAVs searching the environment, the upper bound of the search time t_s is proven to be Eq. (18). □

This proposition provides only a worst case bound, which is far from tight. The simulation results presented in Sec. 8 show that using the proposed strategy, the performance is much better than what the above upper bound implies.

7 Opportunistic Learning Algorithms

This section discusses the learning algorithms used to estimate the occupancy information. As described earlier, UAV i uses tripartite neural networks for predicting $\nu_{k_i}(t+2)$, each subnetwork predicting the two-step occupancy of one of the target cells. This is accomplished with a Q-learning procedure [55], using the true occupancy values, $\nu_{k_i}(t+2)$, which become available at time $t+2$. Essentially, the neural networks learn to produce an increasingly accurate estimate of $\nu_{k_i}(t+2)$ given $s_i(t)$. The weights of the neural networks are modified using the Levenberg-Marquardt procedure.

We consider three situations for training the neural networks for each UAV i.

Centralized learning. In this case, there is only one tripartite neural network. All UAVs communicate their true observations of $\nu_{k_i}(t+2)$ to this network, which calculates the errors for all its corresponding predictions and uses these for learning. One can say that all the UAVs in the team share the same tripartite neural network as a "brain" and utilize all their experiences to train this brain.

Decentralized learning. In this case, each UAV has its own tripartite neural network, which is trained using its own predictions and observations. There is no experience sharing among the group of UAVs. train independently

Opportunistic Cooperative Learning. In this case, UAVs learn as in the decentralized learning (DL) case, each UAV maintaining a running *reward average*

$$\bar{\rho}_i(t+1) = (1-\alpha_i)\bar{\rho}_i(t) + \alpha_i \rho_i(t+1) \quad 0 < \alpha_i \leq 1$$

which shows how well it has done recently. This is used as a measure of performance for its predictor. When two UAVs find themselves in neighboring cells, they compare their $\bar{\rho}$ values, and the UAV with the lower value copies the network and the $\bar{\rho}$ value of the other UAV with probability π. Note that $\pi=0$ corresponds to the DL case.

The rationale behind this study is as follows. One would expect that in a homogeneous environment, a centralized brain, being trained with information from all UAVs, would be better than several "minibrains," each trained on the limited experience of a single UAV. However, the centralized learning (CL) approach has several drawbacks: (1) Since all UAVs rely on the same neural network, the time to obtain a prediction increases with the number of UAVs. (2) All UAVs are forced to follow the same prediction model, thus precluding the possibility of better models emerging. (3) Constant communication between UAVs and the central network is needed, thus wasting energy. (4) The approach is not robust since error or malfunction in the central network can disrupt the whole system. Also, the CL approach breaks down completely in nonhomogeneous environments. The DL approach, on the other hand, implicitly explores the space of possible predictive models as each UAV builds its own neural network. However, the information used in training each UAV's network is necessarily limited to the UAV's own experience. One possible way around this is to have UAVs occasionally compare their models, and have less successful UAVs adopt the models of more successful ones with some *copying probability*. This is the opportunistic cooperative learning (OCL) approach, so named because UAVs use the opportunity afforded by proximity to improve performance via cooperation. This is essentially a guided stochastic learning model like simulated annealing, particle swarm optimization, and genetic algorithms. In our study, we consider how the copying probability affects the performance of the system relative to the CL and DL methods. While it is possible to envision several models for the copying probability π, for simplicity, we focus only on the case when it is constant.

8 Simulation Results

In this section, we present some simulation results to show the effectiveness of the OCL search strategy. We consider how the copying probability affects the performance of the system relative to the CL and DL methods. We mainly explore two issues in the simulation studies:

- comparing the performance of the CL approach and the DL approach
- determining whether opportunistic copying of superior predictors by unsuccessful UAVs in the DL scenario leads to an improvement in performance and speed of learning

While, in practice, the UAVs would learn as they search, we have used a two-phase approach to evaluate the performance of the various approaches. In the *training phase*, the system is trained over n_{train} steps, with the neural network weights modified at each step. The training is done using a search in an actual environment, but the uncertainty values of visited cells are repeatedly reset to 1 after they are visited a few times, thus creating greater opportunity for learning. The training phase is followed by an *evaluation phase*, during which the trained UAVs search an environment for n_{eval} steps without any further learning. This shows how the uncertainty about the environment is reduced over time with UAVs trained by different algorithms. The results for the search strategy based on the greedy estimation described in Sec. 7 are also plotted for comparison.

A representative simulation study is presented in this section. In this study, we simulated a team of 15 UAVs searching a 20×20 cellular environment. There is no a priori topographical information and no other sources of information about the environment.

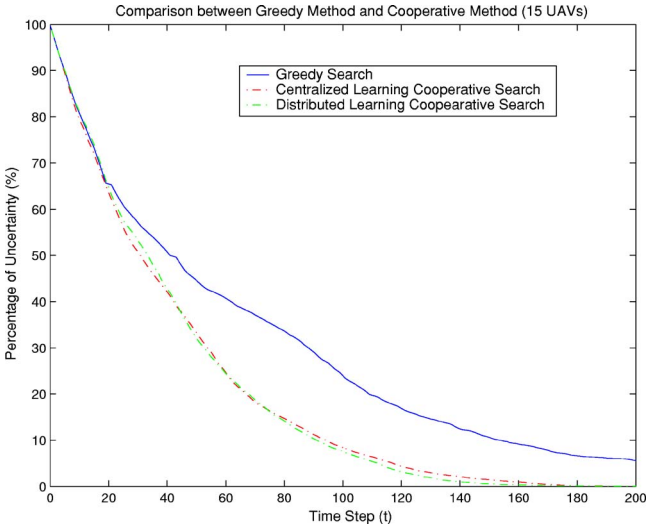

Fig. 5 Search performance for 15 UAVs. The CL and DL algorithms use 100 steps of learning.

Thus, $z(x,y,0)=1$ holds for each cell (x,y) in the environment. For all the simulation runs in this paper, the UAVs start at the same initial locations and with the same orientations, as shown in Figs. 8 and 9. The UAVs' sensor uncertainty reduction rate is set at $\eta=0.5$. Figure 5 shows the time course of uncertainty reduction when 15 UAVs trained with various algorithms search the environment. It is shown that by using the cooperative search strategy developed in this paper, the search time used to reach a 98% certainty over the whole environment is less than 200 steps, which is much more less than the theoretical upper bound of the search time for the group UAVs to fully search the environment to $C=0.02$ obtained from Eq. (18), $1.9756e+034$. It also shows that a greedy search performs much more poorly than the cooperative algorithms. This effect is likely to increase with the number of UAVs since that also enhances the quality of learning. The figure indicates that both cooperative algorithms perform equally well. However, this is the result of the long training time (100 steps).

Figure 6 shows how rapidly uncertainty is reduced by UAVs

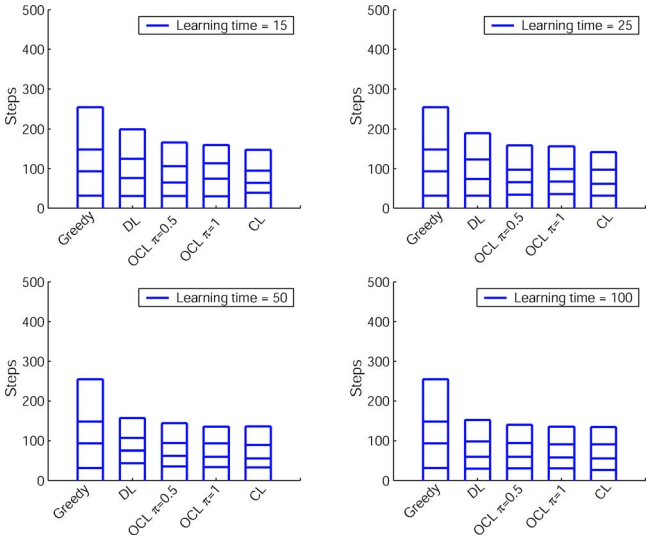

Fig. 6 Search efficiency: The increments on each bar indicate the number of search steps needed to reduce uncertainty by 50%, 75%, 90%, and 98%. The system has 15 UAVs searching a 20×20 environment.

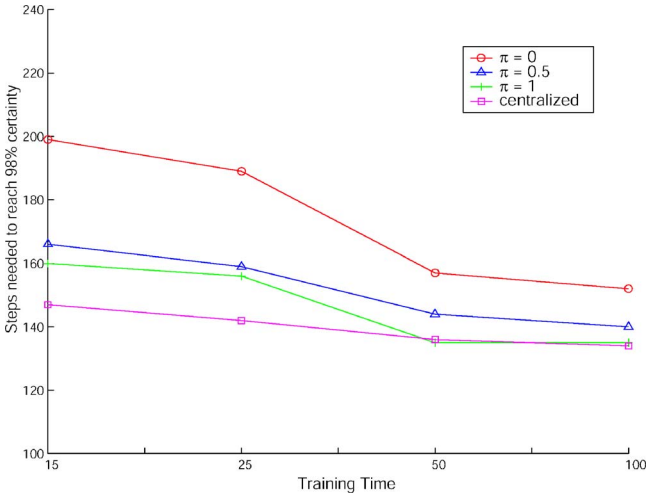

Fig. 7 Search performance for 15 UAVs. The CL and DL algorithms use 100 steps of learning.

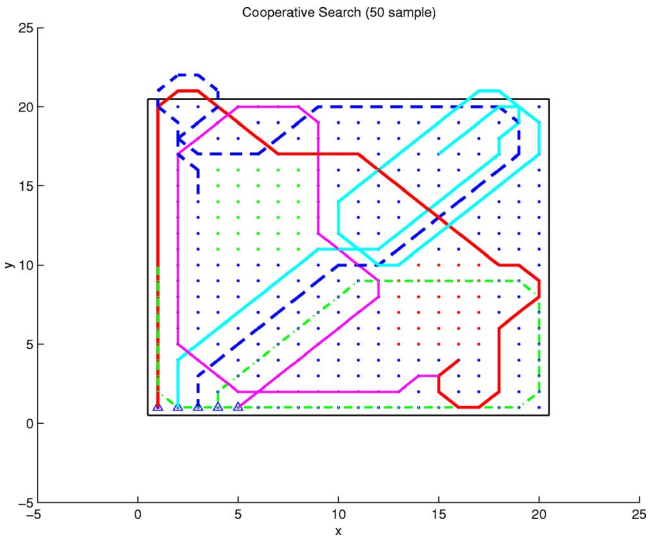

Fig. 9 Search paths for UAVs in a five UAV system using OCL with $\pi=0.5$

trained with different algorithms for 15, 25, 50, and 100 steps. Clearly, the OCL algorithms with $\pi>0$ learn faster than the DL algorithm and almost as fast as the CL algorithm. This is confirmed by Fig. 7, which shows the time needed to reach a 98% certainty. However, it is apparent that OCL needs about 50 steps of learning before reaching the performance of CL, while DL is consistently worse than both.

Finally, Figs. 8 and 9 show the actual search paths taken by five UAVs searching a 20×20 environment. It is apparent that greedy UAVs (Fig. 8) tend to follow each other—essentially "picking the crumbs" left by other UAVs, while cooperative UAVs (Fig. 9) are able to find more diverse search paths.

9 Conclusion

One of the key issues for a successful deployment of networked multiple UAV systems is the design of cooperative decision making and control strategies. In this paper, based on a previously developed multi-UAV cooperative search framework, we mathematically formulate a cooperative search problem using a discretized cellular space and present an intelligent learning method to enable the coordination among the group vehicles.

In the proposed cooperative search model, the UAVs learn to predict the behavior of other UAVs in their neighborhood using neural networks trained via RL. We compared the search performance of CL, where all UAVs use a single centrally trained adaptive predictor, with that of the DL case, where each UAV has its own predictor. Although the centralized approach provided better performance, this came at the cost of efficiency and robustness. To obtain the benefits of both approaches, we proposed the OCL approach, where UAVs learn individually, but also share the benefits of this learning. Because of this cooperation, successful predictors tend to proliferate through the UAV population selectively, without imposing the rigidity of a single centralized predictor. Using simulations, we have shown that the OCL approach can provide a prediction performance close to that of CL while retaining the advantages of DL.

We have shown that teams of UAVs can completely accomplish the search task by following our strategy and given some mathematical analysis of the performance. However, many problems remain open, such as the theoretical analysis of the speedup provided by cooperation and the derivation of a tight upper bound on search time. Research work on these issues is ongoing and will be presented in future reports.

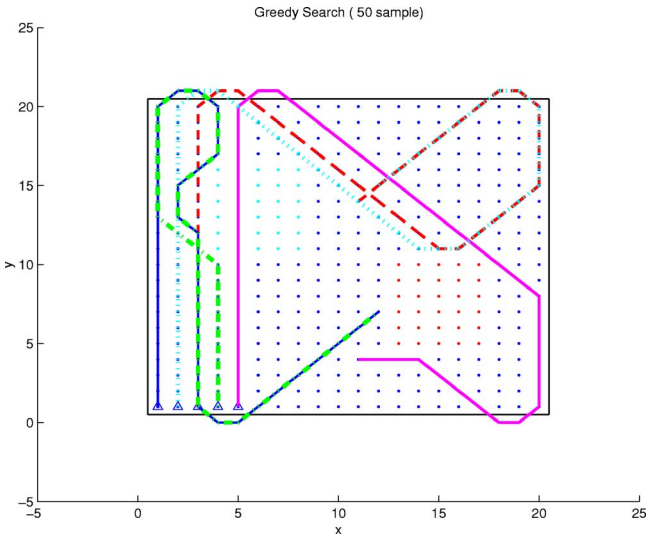

Fig. 8 Search paths for UAVs in a five UAV system using greedy search. Note that many paths overlap, reducing search efficiency.

Acknowledgment

This research was sponsored by the Defense Advanced Research Project Agency (DARPA) under Contract No. F33615-01-C3151 issued by the AFRL/VAK.

Appendix: Derivation of the Uncertainty Update Equation

In this part, we describe how to use the Dempster-Shafer (DS) evidential method as the basic approach for sensor data fusion in building the uncertainty cognitive map, and provide a theoretical explanation of the uncertainty reduction rate defined in Sec. 4.2. To aid the reader, the basics of the DS theory are briefly reviewed first.

A.1 Dempster-Shafer Theory

The DS evidence method is considered a generalized Bayesian theory but has the advantage of being able to clearly distinguish ignorance and contradiction [56]. The basic entity in the DS theory is a set of exclusive and exhaustive hypotheses about some

problem domain. It is called the *frame of discernment*, denoted as Θ. The degree of belief in each hypothesis is represented by a real number in [0,1]. The *basic probability assignment* (BPA) is a function $m:\Psi \rightarrow [0,1]$, where Ψ is the set of all subsets of Θ, the power set of Θ, $\Theta = 2^{\Theta}$. The function $m(\,)$ can be interpreted as distributing belief to each of element in Ψ, with the following criteria satisfied:

$$\sum_{A \in \Psi} m(A) = 1 \tag{A1}$$

$$m(\Phi) = 0 \tag{A2}$$

Thus, the element A is assigned a basic probability number $m(A)$ describing the degree of belief that is committed exclusively to A. Note that a situation of total ignorance is characterized by $m(\Theta) = 1$. The total evidence that is attributed to A is the sum of all probability numbers assigned to A and its subsets

$$Bel(A) = \sum_{\forall B: B \subseteq A} m(B) \tag{A3}$$

$Bel(A)$, also called the *credibility* of A, is interpreted as a measure of the total belief committed to A. It can easily be verified that the belief in some hypothesis A and the belief in its negation $\bar{A}$ do not necessarily sum to 1, which is a major difference with the probability theory. The *plausibility* of A is defined to indicate the extent to which the evidence does not support $\bar{A}$ and is defined as

$$Pl(A) = 1 - Bel(\bar{A}) = \sum_{B \cap A \neq \Phi} m(B) \tag{A4}$$

In this formulation, the probability of a proposition A is expressed as an evidential interval

$$[Bel(A) Pl(A)] \tag{A5}$$

For that reason, *Bel* and *Pl* are also called *lower* and *upper* probabilities, respectively. The above interval reduces to a point in the case of a Bayesian belief function.

Given two belief functions over the same frame of discernment but induced by two independent sources of information, they can be combined into a new belief function over that frame of discernment using Dempster's rule of combination

$$m_1 \oplus m_2(A) = \frac{\sum_{B \cap C = A} m_1(B) m_2(C)}{1 - \sum_{B \cap C = \Phi} m_1(B) m_2(C)} \tag{A6}$$

$$m_1 \oplus m_2(\Phi) = 0 \tag{A7}$$

A.2 A General Dempster-Shafer Theory Based Map-Building Algorithm

In the search problem, the group of UAVs search the environment to gather the target distribution information in the environment. Therefore, a cell in the environment is characterized by two states, *empty* and *full*. To use Dempster's rule, we define the field of discernment Θ by the set

$$\Theta = \{E, F\} \tag{A8}$$

where the E and F correspond to the possibilities that the cell is empty (no target present denoted as $s(x,y)=0$) or full (target present denoted as $s(x,y)=1$), respectively. The set of all subsets of Θ is the power set

$$\Lambda = 2^{\Theta} = \{\Phi, E, F, U\} \tag{A9}$$

where $U=\{E,F\}$ represents the *unknown*. The state of cell (x,y) is described by assigning basic probability numbers to each element in Λ satisfying

$$m_{x,y}(\Phi) = 0 \tag{A10}$$

$$\sum_{A \in \Lambda} m_{x,y}(A) = m_{x,y}(E) + m_{x,y}(F) + m_{x,y}(U) = 1 \tag{A11}$$

Considering this linear dependence, it is sufficient to store $m_{x,y}(F)$ and $m_{x,y}(U)$ to represent the state of cell (x,y). The cognitive map used to store $m_{x,y}(F)$ is called the *target map*, denoted as $F(t)$, and the map to store $m_{x,y}(U)$ is called the *ignorance map*, denoted as $U(t)$. The cell (x,y) is initialized as $m_{x,y}(F)=m_{x,y}(E)=0$ and $m_{x,y}(U)=1$ if there is no available a priori information on targets, representing total ignorance about the state of that cell. Each sensor reading obtained during the UAVs' search of the environment is a source of evidence about the state of the cell, and it can be fused into maps through the sensor models.

The sensor model converts the sensor readings into belief assignments and can be considered its BPA function. When a sensor reading, denoted by $b(x,y,t)$, reports a target detection in cell (x,y) at time t, this sensor reading can be regarded as a piece of evidence that increases our belief that there is a target present in cell (x,y), i.e., belief in state F. However, this piece of evidence does not by itself provide 100% certainty due to the sensor's inaccuracy. This can be expressed by saying that only some part of our belief is committed to the target present. Since this sensor reading does not provide any information about the state E, the belief in E cannot be changed, and the change must be assigned to U. This item of evidence can therefore be represented by the BPA function defined as

$$m_b(E) = 0 \tag{A12}$$

$$m_b(F) = m_f \tag{A13}$$

$$m_b(U) = 1 - m_f \tag{A14}$$

where m_f denotes our incremental belief in the target's presence given a sensor reporting a target detection.

Similarly, when a sensor reading $b(x,y,t)$ reports no target detection in cell (x,y) at time t, there is no information about the target being present and the BPA function is given as

$$m_b(E) = m_e \tag{A15}$$

$$m_b(F) = 0 \tag{A16}$$

$$m_b(U) = 1 - m_e \tag{A17}$$

where m_e denotes our incremental belief in the target's absence given a sensor reporting no target detection. Note that m_f and m_e are parameters indicating the accuracy of the sensor.

Based on the sensor belief definition, each sensor reading of the environment can be fused into the map using Dempster's rule of combination (Eq. (A6)). By adding subscripts b and x,y to the basic probability masses m, we describe the BPAs of the sensor and the map. Explicitly, the new BPAs for each cell in the map are

$$m_{x,y} \oplus m_b(E) = \frac{m_{x,y}(E) m_b(E) + m_{x,y}(E) m_b(U) + m_{x,y}(U) m_b(E)}{1 - m_{x,y}(E) m_b(F) - m_{x,y}(F) m_b(E)} \tag{A18}$$

$$m_{x,y} \oplus m_b(F) = \frac{m_{x,y}(F) m_b(F) + m_{x,y}(F) m_b(U) + m_{x,y}(U) m_b(F)}{1 - m_{x,y}(E) m_b(F) - m_{x,y}(F) m_b(E)} \tag{A19}$$

$$m_{x,y} \oplus m_b(U) = \frac{m_{x,y}(U) m_b(U)}{1 - m_{x,y}(E) m_b(F) - m_{x,y}(F) m_b(E)} \tag{A20}$$

Equations (A18)–(A20) represent a general function to update the UAVs' knowledge about target distribution based on sensor readings. It can be shown that $m_{x,y}(U)$, which represents ignorance of

the state in a cell (x,y), will decrease as the cell is searched.

A.3 Uncertainty Update Rule Derivation

In this paper, we focus mainly on measuring uncertainty, so whether a sensor reading updates the empty or full state of the cell does not matter. We also assume that all the sensors carried by each vehicle are identical. Therefore, we can arbitrarily use the BPA for positive or negative sensor readings to update the uncertainty after each reading and assume that the change is always by the same fixed value, β. We define a simplified BPA function for a single sensor reading as

$$m_b(E) = 0 \tag{A21}$$

$$m_b(F) = \beta \tag{A22}$$

$$m_b(U) = 1 - \beta \tag{A23}$$

i.e., assuming $m_f = \beta$, and use this BPA in the update function (Eqs. (A18)–(A20)). Given that $m_{x,y}(E) = 0$ at $t = 0$ and the simplified BPA leaves it unchanged, both the subtractive terms in the denominator of Eq. (A20) equal zero, giving the simplified update rule,

$$m_{x,y} \oplus m_b(U) = (1 - \beta) m_{x,y}(U) \tag{A24}$$

By identifying $m_{x,y} \oplus m_b(U)$, $m_{x,y}(U)$, and $1-\beta$ with $z(x,y,t+1)$, $z(x,y,t)$, and η, respectively, we have the following update rule for updating the uncertainty value of cell (x,y) after each reading:

$$z(x,y,t+1) = \eta z(x,y,t) \tag{A25}$$

This update rule is the same as the uncertainty update rule denoted by Eq. (2). This shows that the uncertainty model used in this paper corresponds to a DS map-building algorithm using a specific simplified case. Note that a simplified BPA using the negative sensor reading case would have produced the same uncertainty update rule.

References

[1] Schoenwald, D. A., 2000, "AUVs: In Space, Air, Water, and on the Ground," IEEE Control Syst. Mag., **20**(6), pp. 15–18.

[2] Office of The Under Secretary of Defense for Acquisition Technology and Logistics Washington DC, 2002, "Unmanned Aerial Vehicles Roadmap 2002–2027," Office of the Secretary of Defense Technical Report A809414; http://www.acq.osd.mil/usd/uav-roadmap.pdf

[3] Chandler, P., Rasmussen, S., and Pachter, M., 2000, "UAV Cooperative Path Planning," *Proceedings of AIAA Guidance, Navigation, and Control Conference and Exhibit*, Denver, CO, pp. 1255–1265.

[4] Chandler, P., and Pachter, M., 2001, "Hierarchical Control for Autonomous Teams," *Proceedings of AIAA Guidance, Navigation, and Control Conference and Exhibit*, Monterey, CA, pp. 632–642.

[5] Pachter, M., and Chandler, P., 1998 "Challenges of Autonomous Control," IEEE Control Syst. Mag., **18**(4), pp. 92–97.

[6] Nygard, K., Chandler, P., and Pachter, M., 2001, "Dynamic Network Optimization Models for Air Vehicle Resource Allocation," *Proceedings of the 2001 American Control Conference*, Arlington, VA, Vol. 3, pp. 1853–1856.

[7] Jin, Y., Minai, A. A., and Polycarpou, M. M., 2003, "Cooperative Real-Time Search and Task Allocation in UAV Teams," *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui, HI, Vol. 1, pp. 7–12.

[8] Chandler, P., 2001, "UAV Cooperative Classification," *Workshop on Cooperative Control and Optimization*, Kluwer Academic, Dordrecht, pp. 1–20.

[9] Bellingham, J. S., Tillerson, M., Alighanbari, M., and How, J. P., 2002, "Cooperative Path Planning for Multiple UAVs in Dynamic and Uncertain Environment," *Proceedings of the 41st IEEE Conference on Decision and Control*, Las Vegas, NV, Vol. 9, pp. 2816–2822.

[10] Yang, Y., Polycarpou, M., and Minai, A., 2004, "Decentralized Cooperative Search by Networked Multi-UAVs in Uncertain Environment," *Proceedings of the 2004 American Control Conference*, Boston, MI, Vol. 6, pp. 5558–5563.

[11] Flint, M., Polycarpou, M., and Fernandez, E., 2003, "Stochastic Models of a Cooperative Autonomous UAV Search Problem," Military Operations Research Journal, **8**(4), pp. 13–32.

[12] Vincent, P., and Rubin, I., 2004, "A Framework and Analysis for Cooperative Search Using UAV Swarms," *Proceedings of the 2004 ACM Symposium of Applied Computing*, Nicosia, Cyprus, Vol. 1, pp. 79–86.

[13] McLain, T., Chandler, P., Rasmussen, S., and Pachter, M., 2001, "Cooperative Control of UAV Rendezvous," *Proceedings of the 2001 American Control Conference*, Arlington, VA, Vol. 3, pp. 2309–2314.

[14] Tanner, H., Jadbabaie, A., and Pappas, G. J., 2003, "Stable Flocking of Mobile Agents, Part I: Fixed Topology," *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui, HI, Vol. 2, pp. 2010–2015.

[15] Tanner, H., Jadbabaie, A., and Pappas, G. J., 2003, "Stable Flocking of Mobile Agents, Part II: Dynamic Topology," *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui, HI, Vol. 2, pp. 2016–2021.

[16] Polycarpou, M., Yang, Y., Liu, L., and Passino, K., 2003, "Cooperative Control Design for Uninhabited Air Vehicles," *Cooperative Control: Models, Applications and Algorithms*, Kluwer Academic, Dordrecht, pp. 283–321.

[17] Polycarpou, M., Yang, Y., and Passino, K., 2001, "A Cooperative Search Framework for Distributed Agents," *Proceedings of the 2001 IEEE International Symposium on Intelligent Control*, Mexico City, Mexico, pp. 1–6.

[18] Tan, M., 1993, "Multi-Agent Reinforcement Learning: Independent Versus Cooperative Agents," *Proceedings of the Tenth International Conference on Machine Learning*, pp. 330–337.

[19] Koopman, B., 1980, *Search and Screening: General Principles With Historical Application*, Pergamon, New York.

[20] Stone, L., 1995, *Theory of Optimal Search*, Academic, New York.

[21] Eagle, J., and Yee, J., 1990, "An Optimal Branch-and-Bound Procedure for the Constrained Path Moving Target Search Problem," Oper. Res., **38**(1), pp. 110–114.

[22] Hohzaki, R., and Iida, K., 1995, "Path Constrained Search Problem With Reward Criterion," J. Oper. Res. Soc. Jpn., **38**(2), pp. 254–264.

[23] Hohzaki, R., and Iida, K., 1995, "An Optimal Search Plan for a Moving Target When a Search Path is Given," Math. Japonica, **41**(1), pp. 175–184.

[24] Benkoski, S., Monticino, M., and Weisinger, J., 1991, "A Survey of the Search Theory Literature," Naval Res. Logistics Quart., **38**, pp. 469–494.

[25] Richardson, H., 1987, in *Search Theory: Some Recent Developments*, D. Chudnovsky and G. Chudnovsky, eds., Marcel Dekker, New York, pp. 1–12.

[26] Chandler, P., Pachter, M., and Rasmussen, S., 2001, "UAV Cooperative Control," *Proceedings of the 2001 American Control Conference*, Arlington, VA, Vol. 1, pp. 50–55.

[27] Bellingham, J. S., Tillerson, M., Richards, A. G., and How, J. P., 2001, "Multi-Task Assignment and Path Planning for Cooperating UAVs," *Conference on Cooperative Control and Optimization*, Kluwer Academic, Dordrecht, pp. 1–19.

[28] Jin, Y., Minai, A. A., and Polycarpou, M. M., 2004, "Balancing Search and Target Response in Cooperative UAV Team," *Proceedings of the 43rd IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, Vol. 3, pp. 2923–2928.

[29] Murphey, R. A., 1999, "An Approximate Algorithm for a Weapon Target Assignment Stochastic Program," *Approximate and Complexity in Numerical Optimization: Continuous and Discrete Problem*, Kluwer Academic, Dordrecht.

[30] Burgard, W., Fox, D., Moors, M., Simmons, R., and Thrun, S., 2000, "Collaborative Multi-Robot Exploration," *Proceedings of International Conference on Robotics and Automation*, Vol. 1, pp. 476–481.

[31] Choset, H., and Pignon, P., 1997, "Coverage Path Planning: The Boustrophedon Cellular Decomposition," *International Conference on Field and Service Robotics*, Canberra, Australia.

[32] Rekleitis, I., Dudek, G., and Milios, E., 1998, "Accurate Mapping of an Unknown World and Online Landmark Positioning," *Proceedings of Vision Interface*, pp. 455–461.

[33] Svennebring, J., and Koenig, S., 2004, "Building Terrain-Covering Ant Robots: A Feasibility Study," Auton. Rob., **16**(3), pp. 313–332.

[34] Wagner, I. A., Lindenbaum, M., and Bruckstein, A. M., 2001, "MAC Versus PC: Determinism and Randomness as Complementary Approaches to Robotic Exploration of Continuous Unknown Domains," Int. J. Robot. Res., **19**(1), pp. 313–332.

[35] Yang, S., and Luo, C., 2004, "A Neural Network Approach to Complete Coverage Path Planning," IEEE Trans. Syst., Man, Cybern., Part B: Cybern., **34**(1), pp. 718–725.

[36] Choset, H., 2001, "Coverage for Robotics—A Survey of Recent Results," Ann. Math. Artif. Intell., **31**(1–4), pp. 113–126.

[37] Yang, Y., Polycarpou, M., and Minai, A., 2002, "Opportunistically Cooperative Neural Learning in Mobile Agents," *Proceedings of the 2002 International Joint Conference on Neural Networks*, Honolulu, HI, Vol. 3, pp. 2638–2643.

[38] Yang, Y., Minai, A., and Polycarpou, M., 2002, "Decentralized Cooperative Search in UAV's Using Opportunistic Learning," *Proceedings of AIAA Guidance, Navigation, and Control Conference and Exhibit*, Monterey, CA, Paper No. AIAA-2002-4590.

[39] Yang, Y., Minai, A., and Polycarpou, M., 2002, "Analysis of Opportunistic Method for Cooperative Search by Mobile Agents," *Proceedings of the 41st IEEE Conference on Decision and Control*, Las Vegas, NV, Vol. 1, pp. 576–577.

[40] Yang, Y., Polycarpou, M., and Minai, A., 2005, "Evidential Map-Building Approaches for Multi-UAV Cooperative Search," *Proceedings of the 2005 American Control Conference*, Portland, OR, pp. 116–121.

[41] Flint, M., Fernandez, E., and Polycarpou, M., 2003, "Cooperative Control for UAVs Searching Risky Environments for Targets," *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui, HI, Vol. 4, pp. 3567-–3572.

[42] Flint, M., Polycarpou, M., and Fernandez, E., 2002, "Cooperative Control for Multiple Autonomous UAV's Searching for Targets," *Proceedings of the 41st IEEE Conference on Decision and Control*, Las Vegas, NV, Vol. 3, pp. 2823–2828.

[43] Flint, M., Polycarpou, M., and Fernandez, E., 2002, "Cooperative Path-Planning for Autonomous Vehicles Using Dynamic Programming," *Proceedings of the IFAC 15th Triennial World Congress*, Barcelona, Spain, pp. 1694–1699.

[44] Sujit, P. B., and Ghose, D., 2004, "Multiple Agent Search of an Unknown Environment Using Game Theoretical Models," IEEE Trans. Aerosp. Electron. Syst., **40**(1–4), pp. 491–508.
[45] Sujit, P. B., and Ghose, D., 2003, "Optimal Uncertainty Reduction Search Using the *k*-Shortest Path Algorithm," *Proceedings of the 2003 American Control Conference*, Denver, CO, Vol. 3, pp. 3269–3274.
[46] Enns, D., Bugajski, D., and Pratt, S., 2002, "Guidance and Control for Cooperative Search," *Proceedings of the 2002 American Control Conference*, Anchorage, AK, Vol. 3, pp. 1923–1929.
[47] Sujit, P. B., and Ghose, D., 2004, "Multiple Agent Search of an Unknown Environment Using Game Theoretical Models," *Proceedings of the 2004 American Control Conference*, Boston, MI, Vol. 6, pp. 5564–5569.
[48] Baum, M. L., and Passino, K. M., 2002, "A Search-Theoretic Approach to Cooperative Control for Uninhabited Air Vehicles," *Proceedings of AIAA Guidance, Navigation, and Control Conference and Exhibit*, Monterey, CA, Paper No. AIAA-2002-4589.
[49] Zhang, C., and Ordonez, R., 2003, "Decentralized Adaptive Coordination and Control of Uninhabited Autonomous Vehicles Via Surrogate Optimization," *Proceedings of the 2003 American Control Conference*, Vol. 3, pp. 2205–2210.
[50] Ablavsky, V., and Snorrason, M., 2000, "Optimal Search for a Moving Target: A Geometric Approach," *Proceedings of AIAA Guidance, Navigation, and Control Conference and Exhibit*, Denver, CO, pp. 14–17.
[51] Tirumalai, A. P., Schunck, B. G., and Jain, R. C., 1995, "Evidential Reasoning for Building Environment Maps," IEEE Trans. Syst. Man Cybern., **25**(1), pp. 10–20.
[52] Bertsekas, D. P., 1995, *Dynamic Programming and Optimal Control*, Athena Scientific, MA, Vol. 1.
[53] Dubins, L., 1957, "On Curves of Minimal Length With a Constraint on Average Curvature and With Prescribed," Am. J. Math., **79**(3), pp. 497–516.
[54] Wagner, I. A., Lindenbaum, M., and Bruckstein, A. M., 1999, "Distributed Covering by Ant-Robots Using Evaporating Traces," IEEE Trans. Rob. Autom., **15**(5), pp. 918–933.
[55] Watkins, C. J., 1989, "Learning With Delayed Rewards," Ph.D thesis, University of Cambridge.
[56] Pagac, D., Nebot, E. M., and Durrant-Whyte, H., 1998, "An Evidential Approach to Map-Building for Autonomous Vehicles," IEEE Trans. Rob. Autom., **14**(v), pp. 623–629.