

## Research Article

# Cooperative Search by Multiple Unmanned Aerial Vehicles in a Nonconvex Environment

**Xiaoting Ji, Xiangke Wang, Yifeng Niu, and Lincheng Shen**

*College of Mechatronic Engineering and Automation, National University of Defense Technology, Changsha, Hunan 410073, China*

Correspondence should be addressed to Yifeng Niu; niuyifeng@nudt.edu.cn

Received 12 January 2015; Revised 18 July 2015; Accepted 26 July 2015

Academic Editor: Alain Vande Wouwer

Copyright © 2015 Xiaoting Ji et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a distributed cooperative search algorithm for multiple unmanned aerial vehicles (UAVs) with limited sensing and communication capabilities in a nonconvex environment. The objective is to control multiple UAVs to find several unknown targets deployed in a given region, while minimizing the expected search time and avoiding obstacles. First, an asynchronous distributed cooperative search framework is proposed by integrating the information update into the coverage control scheme. And an adaptive density function is designed based on the real-time updated probability map and uncertainty map, which can balance target detection and environment exploration. Second, in order to handle nonconvex environment with arbitrary obstacles, a new transformation method is proposed to transform the cooperative search problem in the nonconvex region into an equivalent one in the convex region. Furthermore, a control strategy for cooperative search is proposed to plan feasible trajectories for UAVs under the kinematic constraints, and the convergence is proved by LaSalle's invariance principle. Finally, by simulation results, it can be seen that our proposed algorithm is effective to handle the search problem in the nonconvex environment and efficient to find targets in shorter time compared with other algorithms.

## 1. Introduction

Over the past decade, unmanned air vehicles (UAVs) with functional diversity and low cost have been extensively employed in many civil and military applications, such as environment surveillance, battle reconnaissance, and search and rescue in the hazardous environment [1–3]. With the development of advanced sensing and information processing technology, cooperative search has been one of the most popular utilizations of UAVs equipped with sensors (such as camera, Lidar, and sonar) [4]. The goal of cooperative search is to control multiple UAVs to find several unknown targets deployed in a given region, while maximizing the detection probability and minimizing the expected search time.

The problem of cooperative search with multiple UAVs has been studied extensively due to its critical importance for a myriad of applications [3, 5, 6]. Existing methods can be classified into two categories: predefined flight paths based search and dynamic path planning based search. The former is to generate the flight paths (e.g., parallel lines or outward

spirals) in advance and follow these paths during search execution. The typical method of this category is sweep-line based search [7], in which the agents sweep all the points in the given area to find the targets. This method is effective so that no search areas are missed but not efficient due to the predefined paths and cannot be used for searching the dynamic targets. The latter method is to convert the cooperative search problem into a multiagent path planning issue. Typically, the path planning problem is formulated as the optimization of a team objective function subject to a set of constraints [8]. Therefore, dynamic programming [9], artificial intelligence [5, 6], and model predictive control (MPC) [10] can be used for solving these problems. In this paper, inspired by the coverage control, a distributed cooperative search method is proposed by integrating information update into the coverage control scheme, which also belongs to path planning based search methods. In recent years, with the development of related theory in Mobile Wireless Sensor Network (MWSN), coverage control methods [11] have been widely used in multiple robots system. The purpose

of coverage control is to optimally deploy the mobile sensors to maximize the coverage of environment. Due to the limited Field of View (FoV), the cooperative search problem with multiple UAVs can be treated as an optimal coverage problem with a bounded sensing size of sensors [12]. And the distributed control strategy can be easily obtained by optimizing the objective function through the Lloyd algorithm [11]. Besides, by using the coverage control scheme in the search problem, it will not only promote the detection performance of target search but also take account of coverage performance. However, few of the existing coverage control methods have considered the information update about target existence probability, which is crucial to the search problem in practice and essentially affects the movements of UAVs.

Therefore, the first issue that should be addressed is how to integrate the information update into the coverage control scheme in order to solve the search problem. To our knowledge, there are only a few works that utilize the coverage control method with consideration of information update. Zhong and Cassandras [13] use coverage control and data collection to maximize the joint detection probability of random events in a given mission space. However, their goal is to find the optimal deployments for detection, which is essentially different from the search problem. And the density function (importance level of each cell in the region) in their objective formulation depends on the distance to the critical points (i.e., the detected sources). Mirzaei et al. [14] use two different types of vehicles for search and coverage tasks. Search vehicles use a limited look-ahead dynamic programming algorithm to maximize the amount of information gathered by the whole team, while service vehicles use the coverage control method to spread out over the environment to optimally cover the terrain. However, the search and coverage control are executed on different platforms by different schemes, and only the probability distribution of critical points is utilized in the objective function. Besides, the information fusion update is not considered. Hu et al. [15] formulate the path planning as a coverage control problem to find an optimal configuration of all agents that minimizes a given coverage performance cost function. They use the uncertainty information as the density function to cover the region uniformly, which cannot directly facilitate agents to search the region with high target existence. In this paper, we integrate the information update into the coverage control scheme by introducing the probability map of target existence together with the uncertainty map into the density function of coverage optimization. In most coverage control problems, the density function is fixed [16, 17] or only depends on the distance to some critical points [18, 19]. However, our new density function can be adaptively changed, depending on the real-time updated target existence probability and uncertainty map. The higher the probability map is, the larger the probability of target existence is. So utilizing the probability map as the density function can help UAV converge to the regions with high target existence and find targets in a short time. If all the regions with high target existence probability have been searched, or the UAV gets stuck in the local optimum, the uncertainty map can help

the UAV escape from the local optimum and explore more regions.

Due to practical requirements, realistic search areas may be arbitrarily shaped with arbitrary obstacles. Thus, the other issue in our search problem is how to handle the nonconvex environment during search execution. Pimenta et al. [12] use the geodesic distance based algorithm to solve the coverage problem with a network of heterogeneous robots in the nonconvex environment. However, it may cause the robot to get stuck in a saddle point, or even worse, to drive into an obstacle, as the projections of the gradients into target direction of the geodesic distance do not add up to zero when reaching the boundary. Teraoka et al. [19] and Adibi et al. [20] present the potential field-based approaches representing control under the influence of virtual force generated by goals (attraction) and obstacles (repulsion), which have advantages in less computation and disadvantages in being easily trapped into a local minimum of the potential field. Breitenmoser et al. [18] present an algorithm which combines classical Voronoi coverage with Lloyd algorithm and the local planning algorithm TangentBug to compute the motion of the robots around obstacles and corners. It assumes that the range of sensor is infinite, meaning that the sensors can “see” through obstacles, which is unrealistic in practice. Caicedo et al. [16, 17] map a class of connected regions with holes to an almost convex region through a diffeomorphism, such that Voronoi partition and Lloyd algorithm [11] can be used to solve the coverage problem. However, the obstacles in their work must be simple convex ones, and the kinematic model of agents is not considered. As mentioned above, cooperative search in the nonconvex environment with arbitrary obstacles is still an open problem, especially when taking more realistic detection and the kinematic model into consideration. Motivated by this, we propose a generalized method to construct a transformation that can transform the nonconvex region with arbitrary obstacles into an almost convex region. Combining with the asynchronous distributed cooperative search framework, the cooperative search problem in the complex nonconvex environment can be solved.

The main contributions of this work are as follows. First, an asynchronous distributed cooperative search framework is developed that integrates information update into coverage control scheme. And an adaptive density function is formulated depending on real-time updated probability map and uncertainty map, which can balance target search and environment exploration. Second, by extending the diffeomorphism [16], we propose a new transformation method to handle the nonconvex environment with arbitrary obstacles, not limited to convex ones. Based on the transformation, the cooperative search algorithm is provided that can address search problem in the nonconvex environment with arbitrary obstacles. Finally, a control strategy is designed considering the kinematic constraints of UAV, and the convergence is proved using LaSalle’s invariance principle.

The remainder of this paper is organized as follows. In Section 2, some useful preliminaries are provided. The cooperative search framework and the objective formulation,

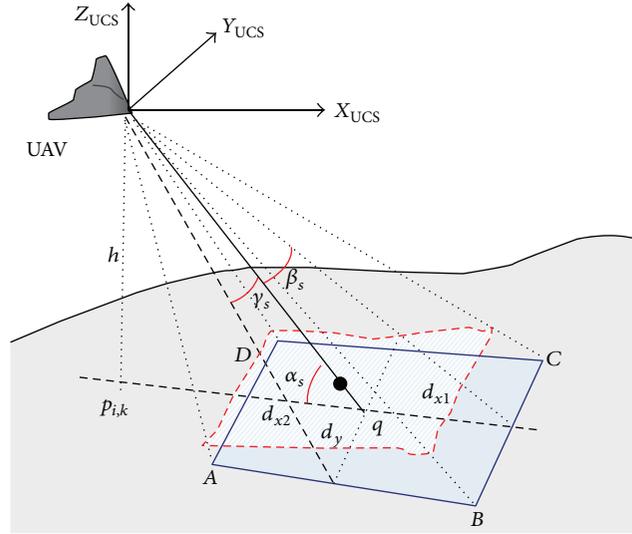


FIGURE 1: FoV of the UAV.

as well as the explicit information update procedure, are presented in Section 3. In Section 4, the generalized method for constructing a transformation is presented to handle the nonconvex environment with arbitrary obstacles. Then the cooperative search algorithm in the nonconvex environment is proposed, and the stability and convergence are proved. Simulation results with analysis are shown in Section 5, and the conclusions are drawn in Section 6.

## 2. Preliminaries

**2.1. Basic Notions and Definitions.** Let  $\mathbb{R}$  be the set of real numbers and let  $\mathbb{R}^n$  denote the  $n$ -dimensional Euclidean space. For a given set  $Q \subseteq \mathbb{R}^n$ ,  $Q^{\text{int}}$  and  $\partial Q$  denote the interior and the boundary of  $Q$ , respectively. One has  $Q^{\text{int}} = Q/\partial Q$ .  $d(x, y) = \|x - y\|$  represents the Euclidean distance between points  $x, y \in \mathbb{R}^2$ , and  $B(q, \varepsilon)$  is a closed ball centered at  $q$  with radius  $\varepsilon$ . In the following, some definitions about the environment are presented.

**Definition 1** (connected partition [21]). Let  $Q \subset \mathbb{R}^2$  be a finite set and let  $P = P_1, \dots, P_N$  be a subset of  $Q$ ; if  $P_i \cap P_j = \emptyset, \forall i \neq j$ , and  $\bigcup_{i=1}^N P_i = Q$ ,  $P = P_1, \dots, P_N$  is said to be a connected partition of  $Q$ .

**Definition 2** (Voronoi partition). A connected partition  $V = \{V_1, \dots, V_N\}$  of  $Q$  is said to be a Voronoi partition of  $Q$ , generated by a vector of distinct points  $c \in Q^N$ , if for each  $V_i$  there exists  $c_i \in V_i$  that satisfies  $d(c_i, q) \leq d(c_j, q), \forall q \in V_i, \forall j \neq i$ .

**Definition 3** (nonconvex [22]). Given any finite set  $Q \subset \mathbb{R}^2$ , point  $q$  of  $\partial Q$  is nonconvex if, for all  $\varepsilon > 0$ , there exist  $q_1$  and  $q_2$  in  $B(q, \varepsilon) \cap \partial Q$  so that the open interval  $(q_1, q_2)$  is outside  $Q$ .

**Definition 4** (nonconvex allowable environment [22]). Let  $Q \subset \mathbb{R}^2$  be a finite set that is allowable if

- (i)  $Q$  is compact and connected;
- (ii)  $\partial Q$  is continuously differentiable except on a finite number of points;
- (iii)  $\partial Q$  has a finite number of nonconvex points which are either isolated points or arcs continuously differentiable everywhere except, possibly, at their end points.

In this paper, the search environment is denoted by  $Q$ , which is a polygon in  $\mathbb{R}^2$ . It is assumed that there are several polygonal obstacles  $H_l \subset Q, l = 1, \dots, M$  within  $Q$ . Thus, the overall feasible region is  $F = Q \setminus \bigcup_{l=1}^M H_l$ . Obviously, the search region is nonconvex since it contains several obstacles.

**2.2. Detection and Communication Model.** Considering  $N$  UAVs in the search region, each UAV is modeled as a nonholonomic point mass moving at a limited speed with a minimum turning radius. The position of UAV  $i$  is denoted by  $p_i = [x_i, y_i]^T$ , and the heading angle is  $\theta_i$ . So the state of UAV  $i$  can be represented as  $[x_i, y_i, \theta_i]$ .

Each UAV is equipped with a camera to take measurements within its Field of View (FoV). Supposing UAV  $i$  is aiming at point  $q$  at time  $k$  (Figure 1), the formulation of FoV is

$$S_{i,k} = \{(x, y) \mid q_x - d_{x1} \leq x \leq q_x + d_{x2}, q_y - d_y \leq y \leq q_y + d_y\},$$

$$\begin{aligned}
q_x &= x_i + \frac{h \cos \theta_i}{\tan \alpha_s}, \\
q_y &= y_i + \frac{h \sin \theta_i}{\tan \alpha_s}, \\
d_{x1} &= h \left( \frac{1}{\tan \alpha_s} - \frac{1}{\tan(\alpha_s + \beta_s)} \right), \\
d_{x2} &= h \left( \frac{1}{\tan(\alpha_s - \beta_s)} - \frac{1}{\tan \alpha_s} \right), \\
d_y &= h \frac{\tan \gamma_s}{\sin \alpha_s},
\end{aligned} \tag{1}$$

where  $h \leq h_{\max}$  represents the flight height of UAV and  $\alpha_s, \beta_s$ , and  $\gamma_s$  are the installed angle and view angles of the camera, related to the physical performance.

The search region  $Q$  is then discretized into several cells, defined as  $o_i$ .  $Z_{o_i,k}$  represents that a target exists in cell  $o_i$  at time  $k$  and  $\bar{Z}_{o_i,k}$  represents that no target exists in cell  $o_i$  at time  $k$ . Similarly,  $D_{o_i,k}$  represents that the UAV detects a target in cell  $o_i$  at time  $k$  and  $\bar{D}_{o_i,k}$  represents that the UAV detects no target in cell  $o_i$  at time  $k$ . Here, we only consider the influence of the Euclidean distance  $\|p_i - o_i\|$  between UAV  $i$  and the detected cell  $o_i$ . The detection probability for UAV  $i$  can be represented by a monotonically decreasing function as

$$P(D_{o_i,k} | Z_{o_i,k}) = \begin{cases} P_{0i} e^{-\lambda \|p_i - o_i\|^2}, & o_i \in \text{FoV} \\ 0, & o_i \notin \text{FoV}, \end{cases} \tag{2}$$

where  $P_{0i} \in (0, 1]$  is a constant of the sensing performance for UAV  $i$  and  $\lambda$  is a positive constant, set to be 1. All the UAVs are provided with the identical kind of camera sensors, so  $P_{0i} = P_{0j} = P_0$  ( $i, j = 1, \dots, N, i \neq j$ ).

In this paper, the communication range is assumed to be limited. Let  $\text{CN} = (\xi, \nu)$  be an undirected graph, where  $\nu = \{1, 2, \dots, N\}$  and  $\xi = \{(i, j) : i, j \in \nu\}$  represent the vertex set and the edge set, respectively. The communication network for UAV  $i$  at time  $k$  is modeled as  $\text{CN}_{i,k} = (\xi_k, \nu)$ . The edge set is  $\xi_k = \{(i, j) : i, j \in \nu \mid d(p_i, p_j) \leq R_C\}$ , where  $R_C$  is the communication range. Therefore, the set of UAV  $i$ 's neighbors can be defined as  $\zeta_{i,k} = \{j \in \nu \mid (i, j) \in \xi_k\}$ . "Neighbor communication strategy" is considered; that is, only when two UAVs reach into each other's communication range, information exchange may happen. DeLima and Pack [23] have proved that this strategy is more efficient in solving the search problem than that of maintaining a global communication network by restricting UAVs' mobility.

### 3. Cooperative Search Framework

The goal of this paper is to develop an efficient cooperative search method for multiple UAVs to find several unknown targets in a nonconvex environment with arbitrary obstacles. Inspired by the coverage control, an asynchronous distributed

cooperative search framework is proposed by integrating the information update into the coverage control scheme, as shown in Figure 2. The proposed framework will promote the detection performance of target search while taking account of coverage performance when the distribution of targets is uniform. Specifically, there are two interdependent tasks. One is online information update containing local update and fusion update. The other is control strategy generation for cooperative search by optimizing the objective function. These two tasks are closely related through the probability map  $\text{TP}_i$  and uncertainty map  $I_i$ . At first, each UAV takes a measurement  $s_i$  and updates its local probability map  $\text{TP}_i$  through Bayesian update rule, where the inaccuracies and uncertainties in the measurements are taken into consideration. Then, UAVs in the same communication network  $\text{CN}_i$  will exchange the information of the probability map and position, as denoted by the blue lines in Figure 2. The shared information of the probability map is fused by individual UAV to maintain a common view of the search region, and the position information is used to update the Voronoi partition  $V_i$ . The uncertainty map  $I_i$  will be obtained by using both position and probability map. Finally, each UAV generates its control law by optimizing the objective function  $H_i$ . Note that this framework is first designed for the convex environment and can also be used for the nonconvex environment, which will be described in the next section. In the following, the explicit description of the objective function and information update will be presented.

#### 3.1. Objective Function Formulation

**3.1.1. Objective Function and Control Law.** First, the cooperative search method in the convex region is presented, which is fundamental to that in the nonconvex region. At each decision, the UAV should determine the optimal control input based on the current state and environment information. The objective is to find several unknown targets, while maximizing the detection probability and minimizing the expected search time. To this end, the following objective function is proposed to be minimized, which is associated with the Voronoi partition:

$$\begin{aligned}
H(P) &= \int_Q f(\|p_i - r\|) \gamma(r) dr \\
&= \sum_{i=1}^N \int_{V_i} f(\|p_i - r\|) \gamma(r) dr,
\end{aligned} \tag{3}$$

where  $r \in Q$  is the cell of environment,  $V_i$  represents the Voronoi region generated by  $p_i$ , and  $f(\|p_i - r\|)$  is a nondecreasing function. The density function  $\gamma(r) : Q \rightarrow \mathbb{R}^+$  represents the (relative) importance level of each cell in the environment  $Q$ . The more important the cell  $r$  is, the larger the value  $\gamma(r)$  would be. Generally speaking, more attention should be paid to the region with a larger density function.

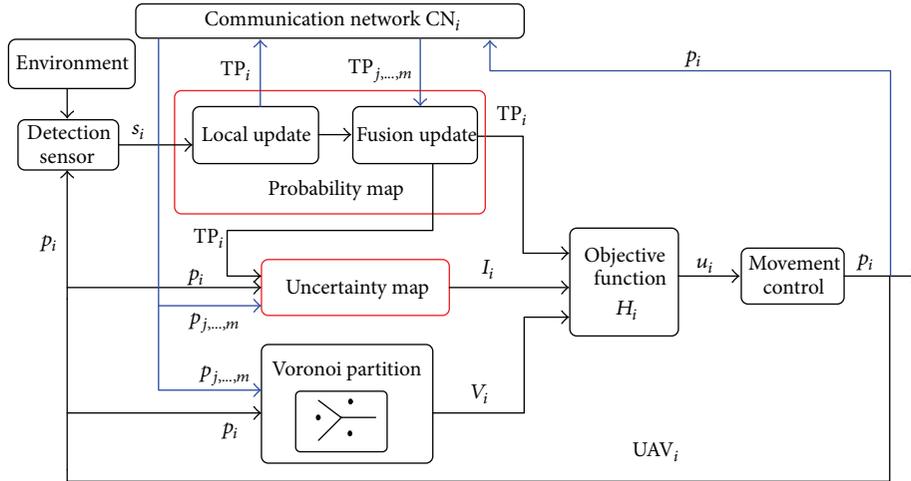


FIGURE 2: Asynchronous distributed cooperative search framework.

In order to optimize the objective function  $H(P)$ , the partial derivative of the objective function should be calculated. Supposing  $f(\|p_i - r\|) = \|p_i - r\|^2$  in this paper, it can be obtained that

$$\begin{aligned} \frac{\partial H}{\partial p_i} &= \frac{\partial H_i(p_i, V_i)}{\partial p_i} = 2 \int_{V_i} (p_i - r) \gamma(r) dr \\ &= 2M_{V_i} (p_i - C_{V_i}), \end{aligned} \quad (4)$$

where  $M_{V_i}$  and  $C_{V_i}$  are the mass and the centroid with respect to the density function  $\gamma(r)$ , defined as

$$\begin{aligned} M_{V_i} &= \int_{V_i} \gamma(r) dr, \\ C_{V_i} &= \frac{1}{M_{V_i}} \int_{V_i} r \gamma(r) dr. \end{aligned} \quad (5)$$

For simplification, the dynamics of each UAV is chosen as  $\dot{p}_i = u_i$ . The gradient descent approach is used to minimize the objective function, expressed as

$$u_i = \dot{p}_i = -k_i (p_i - C_{V_i}), \quad (6)$$

where  $k_i$  is a positive gain.

However, in the nonconvex environment, the above control law may fail for two reasons. The centroid of Voronoi region may lie in the unfeasible area, making it impossible to converge to the centroid. The other is that the trajectory generated by the control law (6) may travel through the unfeasible area. Motivated by this, in the next section, a generalized transform will be presented to transform the nonconvex environment into a convex one such that the above control law can be used.

**3.1.2. Adaptive Density Function.** The density function represents the importance level of each cell in the region, which essentially affects the control strategy of UAVs in the search process. In this paper, we propose an adaptive density function  $\gamma(r)$  that can utilize different types of the information in different search stages. This density function depends on the real-time updated probability map and uncertainty map.

Here, the concept of the uncertainty map  $I_i$  is introduced, defined as the degree of uncertainty about target existence in the environment. The uncertainty map can be derived from the probability map and UAV's trajectories as follows:

$$I_{i,k}(r) = \begin{cases} -TP_{i,k}(r) \ln(TP_{i,k}(r)), & \text{if } TP_{i,k}(r) \neq 0 \\ 0, & \text{if } TP_{i,k}(r) = 0 \text{ and } r \text{ has been detected} \\ 0.5, & \text{if } TP_{i,k}(r) = 0 \text{ and } r \text{ has not been detected.} \end{cases} \quad (7)$$

From the above equations, it can be seen that if the target existence probability is close to the middle section of  $[0, 1]$ , it

will be difficult to determine the existence of the target. Thus, the uncertainty map will have high value in those regions.

Besides, the regions that have not been detected by UAVs also have high values, in order to facilitate covering the unvisited regions. Hence, we define the adaptive density function as follows:

$$\gamma(r) = \begin{cases} TP_{i,k}(r), & \text{Stage } T_1 \\ I_{i,k}(r), & \text{Stage } T_2, \end{cases} \quad (8)$$

where Stage  $T_1$  is from the beginning of search to the time when UAV converges to the centroid of its Voronoi region and gets stuck. From that time on, the second Stage  $T_2$  is started.

This adaptive density function can promote the detection performance of target search while taking account of coverage performance. The probability map is used to lead UAV to move towards the region with high target existence probability, in order to find the targets in a shorter time. However, if the UAV has detected all the regions with high target probability and the centroid of Voronoi region no longer updates, the control law in (6) always leads UAV to the centroid. So the UAV may get stuck in the local optimum. Therefore, the uncertainty map instead of the probability map is used to help UAV escape from the local optimum and explore the regions with large uncertainty.

**3.2. Information Update Method.** In the cooperative search problem, each UAV keeps an individual probability map  $TP_i$ . The information update is in the form of probability map update, and the uncertainty map  $I_i$  is derived from the probability map. The source of the UAV's information consists of three parts: initial information, detection information, and communication information, so the update of individual probability map has two stages: probability map local update and probability map fusion update.

**3.2.1. Probability Map Local Update.** Considering the uncertainty of sensor measurements, the probability map local update is related with whether a target is detected or not, provided that the target existence probability  $P(Z_{o_j,k})$  is known in advance.  $B_{\text{FOV}_i}$  represents that a target is detected by UAV  $i$  at time  $k$  with respect to the sensor measurement  $s_{i,k} = 1$ .  $\bar{B}_{\text{FOV}_i}$  is the opposite of  $B_{\text{FOV}_i}$  with respect to measurement  $s_{i,k} = 0$ . Considering the different sensor measurements, the update processes are different.

(1) If a target is detected by UAV  $i$ , the update of target probability at time  $k$  based on Bayesian theory can be expressed as

$$P(Z_{o_j,k+1} | B_{\text{FOV}_i}) = \frac{P(B_{\text{FOV}_i} | Z_{o_j,k})P(Z_{o_j,k})}{P(B_{\text{FOV}_i})}, \quad (9)$$

$o_j \in Q,$

where  $P(B_{\text{FOV}_i} | Z_{o_j,k})$  is the probability of detecting the target by UAV  $i$ , when a target exists in cell  $o_j$ , and  $P(Z_{o_j,0})$  is the initialized probability map. Since the FoV of UAV  $i$

contains  $M_i$  cells,  $P(B_{\text{FOV}_i} | Z_{o_j,k})$  can be described by the probability of failing to detect the target in these  $M_i$  cells as

$$P(B_{\text{FOV}_i} | Z_{o_j,k}) = 1 - P(\bar{B}_{\text{FOV}_i} | Z_{o_j,k}) = 1 - \prod_{i=1}^{M_i} P(\bar{D}_{o_i,k} | Z_{o_j,k}), \quad (10)$$

where  $P(\bar{D}_{o_i,k} | Z_{o_j,k})$  represents the probability that UAV  $i$  fails to detect the target in cell  $o_i$  when the target is in cell  $o_j$ . Assuming  $p_a$  is false alarm probability,  $P(\bar{D}_{o_i,k} | Z_{o_j,k})$  is described as follows:

$$P(\bar{D}_{o_i,k} | Z_{o_j,k}) = \begin{cases} \bar{p}_a, & o_i \neq o_j \\ \bar{P}_{0i}, & o_i = o_j, \end{cases} \quad (11)$$

where  $\bar{P}_{0i} = 1 - P_{0i}e^{-\lambda_i \|p_i - o_i\|}$  and  $\bar{p}_a = 1 - p_a$ .

If the target is in the FoV of UAV, the detection fault of missing the target occurs in one cell with the probability  $\bar{P}_{0i}$ . If not, the detection result is correct, and only false alarm probability is eliminated. Therefore, the probability for UAV  $i$  of failing to detect the target is

$$\prod_{i=1}^{M_i} P(\bar{D}_{o_i,k} | Z_{o_j,k}) = \begin{cases} \bar{P}_{0i} \times \bar{p}_a^{M_i-1}, & o_j \in S_{i,k} \\ 1 - \bar{p}_a^{M_i}, & o_j \notin S_{i,k}. \end{cases} \quad (12)$$

Combining (10) with (12), it can be obtained that

$$P(B_{\text{FOV}_i} | Z_{o_j,k}) = \begin{cases} 1 - \bar{P}_{0i} \times \bar{p}_a^{M_i-1}, & o_j \in S_{i,k} \\ 1 - \bar{p}_a^{M_i}, & o_j \notin S_{i,k}. \end{cases} \quad (13)$$

Based on the Bayesian complete probability formula,  $P(B_{\text{FOV}_i})$  can be expressed as

$$\begin{aligned} P(B_{\text{FOV}_i}) &= \sum_{o_j \in S_{i,k}} P(B_{\text{FOV}_i} | Z_{o_j,k})P(Z_{o_j,k}) \\ &\quad + \sum_{o_j \notin S_{i,k}} P(B_{\text{FOV}_i} | Z_{o_j,k})P(Z_{o_j,k}) \\ &= \sum_{o_j \in S_{i,k}} (1 - \bar{P}_{0i} \times \bar{p}_a^{M_i-1})P(Z_{o_j,k}) \\ &\quad + \sum_{o_j \notin S_{i,k}} (1 - \bar{p}_a^{M_i})P(Z_{o_j,k}). \end{aligned} \quad (14)$$

Therefore, when the target is detected by UAV  $i$  in its FoV, the update expression of the target existence probability can be obtained as follows:

$$P(Z_{o_j,k} | B_{\text{FOV}_i}) = \begin{cases} \frac{(1 - \bar{P}_{0i} \times \bar{p}_a^{M_i-1})P(Z_{o_j,k})}{P(B_{\text{FOV}_i})}, & o_j \in S_{i,k} \\ \frac{(1 - \bar{p}_a^{M_i})P(Z_{o_j,k})}{P(B_{\text{FOV}_i})}, & o_j \notin S_{i,k}. \end{cases} \quad (15)$$

(2) If the target is not detected by UAV  $i$  in its FoV, the update expression of the target existence probability can be obtained as follows:

$$P(Z_{o_j,k+1} | \bar{B}_{\text{FOV}_i}) = \frac{P(\bar{B}_{\text{FOV}_i} | Z_{o_j,k}) P(Z_{o_j,k})}{P(\bar{B}_{\text{FOV}_i})}, \quad (16)$$

$$o_j \in Q.$$

With the same evolving process as mentioned above, it can be obtained that

$$P(Z_{o_j,k+1} | \bar{B}_{\text{FOV}_i}) = \begin{cases} \frac{\bar{P}_{0i} \times \bar{P}_a^{M_i-1} P(Z_{o_j,k})}{P(\bar{B}_{\text{FOV}_i})}, & o_j \in S_{i,k} \\ \frac{\bar{P}_a^{M_i} P(Z_{o_j,k})}{P(\bar{B}_{\text{FOV}_i})}, & o_j \notin S_{i,k}, \end{cases} \quad (17)$$

where  $P(\bar{B}_{\text{FOV}_i}) = \sum_{o_j \in S_{i,k}} \bar{P}_{0i} \times \bar{P}_a^{M_i-1} P(Z_{o_j,k}) + \sum_{o_j \notin S_{i,k}} \bar{P}_a^{M_i} P(Z_{o_j,k})$ .

As described in (15) and (17), the update of target probability with different sensor measurements at time  $k$  for UAV  $i$  can be summarized as

$$\text{TP}_{i,k+1} = \begin{cases} \frac{(1 - \bar{P}_{0i} \times \bar{P}_a^{M_i-1}) P(Z_{o_j,k})}{P(\bar{B}_{\text{FOV}_i})}, & o_j \in S_{i,k} \text{ and } s_{i,k} = 1 \\ \frac{(1 - \bar{P}_a^{M_i}) P(Z_{o_j,k})}{P(\bar{B}_{\text{FOV}_i})}, & o_j \notin S_{i,k} \text{ and } s_{i,k} = 1 \\ \frac{\bar{P}_{0i} \times \bar{P}_a^{M_i-1} P(Z_{o_j,k})}{P(\bar{B}_{\text{FOV}_i})}, & o_j \in S_{i,k} \text{ and } s_{i,k} = 0 \\ \frac{\bar{P}_a^{M_i} P(Z_{o_j,k})}{P(\bar{B}_{\text{FOV}_i})}, & o_j \notin S_{i,k} \text{ and } s_{i,k} = 0. \end{cases} \quad (18)$$

**3.2.2. Probability Map Fusion Update.** Due to the ‘‘neighbor communication strategy,’’ only when UAV  $i$  and UAV  $j$  fly into each other’s communication range, the information exchange occurs. Each UAV updates its own probability map by fusing information from its neighbors, using the consensus protocol [15]

$$\text{TP}_{i,k} = \sum_{j=1}^N \omega_{i,j,k} \text{TP}_{j,k}, \quad (19)$$

where  $\omega_{i,j,k}$  represents the weight of the target probability map for each UAV, which can be expressed as

$$\omega_{i,j,k} = \begin{cases} 1 - \frac{|\zeta_{i,k}|}{N}, & \text{if } i = j \\ \frac{1}{N}, & \text{if } i \neq j, j \in \zeta_i, k \\ 0, & \text{if } i \neq j, j \notin \zeta_i, k, \end{cases} \quad (20)$$

where  $|\zeta_{i,k}|$  is the number of UAV  $i$ ’s neighbors at time  $k$ .

## 4. Nonconvex Environment with Arbitrary Obstacles

Considering the search environment  $Q \subseteq \mathbb{R}^2$  is nonconvex with arbitrary obstacles  $H$ , we will propose a cooperative search method to address the search problem in the nonconvex environment in this section. Generally, there are two ways to address this problem. One is to find a different objective function that works for nonconvex environments by taking path planning into account, such as the potential field-based method [19, 20]. The second is to find a transformation for the environment [16, 17]. Here, we will choose the second one in order to use the cooperative search framework and formulations in the above section.

Our transformation is inspired by the diffeomorphism idea [16]. However, the diffeomorphism in [16] can only be applied to the nonconvex environment with the convex obstacles. Considering more realistic application, we propose a generalized transformation method that can transform the nonconvex region with arbitrary obstacles (including convex and concave obstacles) into an almost convex region. First, the definition of diffeomorphism is given, as well as the lemma in [16].

*Definition 5* (diffeomorphism [24]). Given two manifolds  $M$  and  $N$ , a differentiable map  $f : M \rightarrow N$  is called a diffeomorphism, if it is a bijection and its inverse  $f^{-1} : N \rightarrow M$  is differentiable as well.

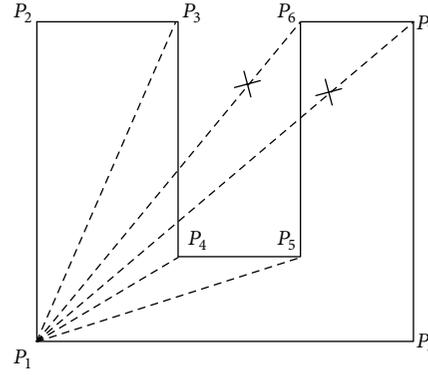
**Lemma 6.** For a given set  $Q \subseteq \mathbb{R}^2$ , let  $\bar{R} \subset \mathbb{R}^2$  be a compact, convex, and simply connected partition of  $Q$  and let  $H \subseteq \bar{R}$  be a convex open set. Let  $\partial H \cap \partial \bar{R} = \emptyset$  and  $R = \bar{R}/H$ . Given  $p' \in H^{\text{int}}$ , there is a.e.  $C^1$  diffeomorphism  $\varphi$  between  $R^{\text{int}}$  and  $\bar{R}^{\text{int}} \setminus \{p'\}$ .

The above lemma follows that  $H$  is convex or star-shaped. However, the real obstacle  $H$  may be arbitrary; the diffeomorphism method in Lemma 6 cannot be used in this case. Therefore, we propose a generalized method to construct the transformation based on a global convex decomposition and some extensions of diffeomorphism in the following.

### 4.1. Diffeomorphism Based Transformation

**4.1.1. Global Convex Decomposition.** As the main preprocessing procedure for constructing the transformation, the global convex decomposition method decomposes the original nonconvex region and the concave obstacles, making it possible to find the diffeomorphism based transformation in the nonconvex environment with concave obstacles.

First, the convex decomposition of concave obstacles is given. Some works have been conducted on the convex decomposition of concave polygons. Schachter [25] utilizes the Delaunay method to decompose polygons into convex sets. Chazelle and Dobkin [26] construct the concave point sets of the  $xk$  type to realize the convex decomposition. Keil [27] uses improved dynamic programming algorithms to achieve decomposing a simple polygon. In this paper,

FIGURE 3: Visible points from  $P_1$ .

our purpose for convex decomposition of concave polygons is to find the diffeomorphism, which is related to the visibility. So a point visibility based convex decomposition method is presented, which can improve the shape quality of the decomposed polygons.

(1) *Find the Concave Points.* Since convex decomposition of concave polygon is to eliminate the concave points, the decomposing line is always drawn from concave points. So the concave points of the concave polygon should be found. The concave point is defined as in Definition 3. The most common method for determining whether a point is concave is to use the vector cross product. Choose the successive points  $p_{i-1}$ ,  $p_i$ , and  $p_{i+1}$  from the point sequence of a polygon in counterclockwise direction. Obtain the vectors  $\overrightarrow{p_{i-1}p_i}$  and  $\overrightarrow{p_i p_{i+1}}$ , and then compute the cross product

$$M = (x_i - x_{i-1}) \times (y_{i+1} - y_i) - (x_{i+1} - x_i) \times (y_i - y_{i-1}), \quad (21)$$

where  $x_i$  and  $y_i$  are the coordinates of point  $p_i$ . If  $M$  is negative, then  $p_i$  is a concave point; otherwise,  $p_i$  is a convex point.

(2) *Find the Visible Point Set.* Point  $q \in Q$  is visible from  $p \in Q$  if  $[p, q] \subset Q$ , where  $[p, q]$  represents the closed path segment  $[p, q] = \{p + \lambda(q - p) \mid \lambda \in [0, 1]\}$ . The visibility set is the set of points in  $Q$  visible from  $p$ . We connect  $P_i$  with other vertices of the polygon in counterclockwise direction. For a given vertex  $P_j$ , if all the rest vertices behind  $P_j$  are on the left side of line  $\overrightarrow{P_i P_j}$ , then  $P_j$  is visible from point  $P_i$ , as shown in Figure 3.

(3) *Find the Decomposing Line Based on the Weighting Function.* The decomposing line is obtained from the concave point to the visible point which is carefully searched from the visible point set of this concave point by using a weight function. In [28], a weight function is defined as  $f(\alpha, \beta) = |\alpha - \beta|$ , where  $\alpha$  and  $\beta$  are the angles between the candidate decomposing line and the neighbor lines. Since  $\alpha, \beta$  lie within the interval  $[0, \pi]$  and the cosine function is monotonous

in this interval, we use the following weight function for computational convenience:

$$f(\alpha, \beta) = |\cos(\alpha) - \cos(\beta)|. \quad (22)$$

The visible point, from which the decomposing line has the minimum weight value, is chosen. For each polygon, repeat the above steps until all the points of the decomposed polygons are convex.

Further, according to the decomposing lines and the edges of the decomposed obstacles, the whole search region is divided into a connected partition  $\tilde{R} = \tilde{R}_1, \dots, \tilde{R}_K$  of  $Q$ . Each partition contains no more than one decomposed convex obstacle. The global decomposition method of the whole region is as follows.

First, extend the decomposing lines from two endpoints to intersect with region boundaries, defined as the dividing lines (for the whole region decomposition), while giving up extending those endpoints that will lead to intersecting with obstacles or other extending lines. Then, from each of those abandoned endpoints, draw a radial line along the two edges of the original concave obstacle to intersect with the region boundary or the obstacle boundary, and obtain the dividing lines. By these dividing lines, the whole region is divided into several subregions, each of which containing no more than one decomposed obstacle. Note that the feasible region of a new subregion may be separated into two parts by the decomposed obstacle. In this case, the subregion is divided again to generate a new small subregion without obstacles and the larger one with the obstacle.

*4.1.2. Extension of Diffeomorphism.* Through convex decomposition, the whole region is divided into several subregions with no more than one convex obstacle. The method for constructing the diffeomorphism in Lemma 6 requires that  $\partial H \cap \partial \tilde{R} = \emptyset$ . However, the new decomposed obstacles may have the common edges with the subregions such that  $\partial H_i \cap \partial \tilde{R}_i \neq \emptyset$ . So the method in Lemma 6 is not appropriate. The expression of the ‘‘common edge’’ means that the edge of the subregion contains the edge of the obstacle. In the following, we will present a generalized method to construct the diffeomorphism for the new decomposed subregions.

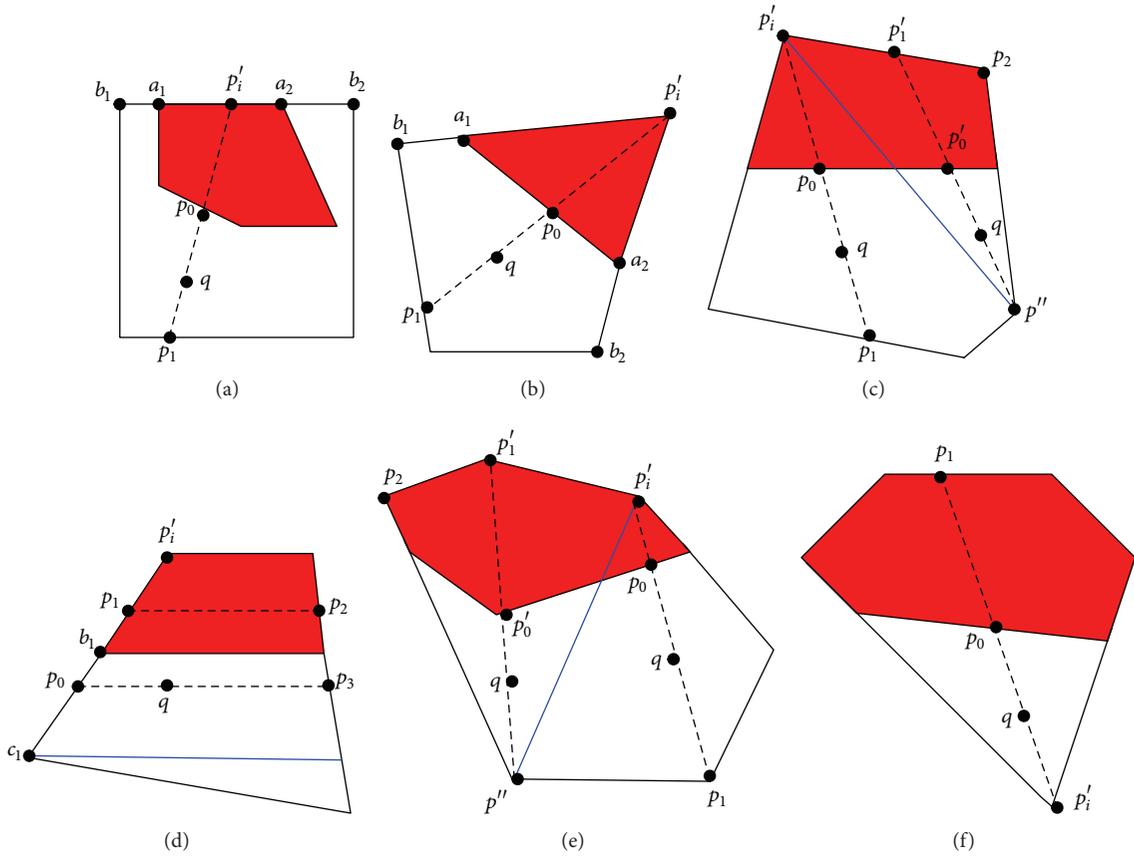


FIGURE 4: Diffeomorphism for the new decomposed regions.

Let  $\tilde{R} = \tilde{R}_1, \dots, \tilde{R}_K$  be a compact, convex, and simply connected partition of  $Q$ , representing the subregions. For each  $\tilde{R}_i$ , let  $H_i \subset \tilde{R}_i$  be a convex open set, representing the decomposed obstacle. Thus, the feasible region is  $R_i = \tilde{R}_i \setminus H_i$ . For this problem, the edges generated by the decomposing lines must be considered. Let  $\partial R_i^{\text{int}} = \partial R_i \setminus (\partial R_i \cap \partial Q)$  be the boundary  $\partial R_i$  but not belong to  $\partial Q$ . Similarly, let  $\partial \tilde{R}_i^{\text{int}} = \partial \tilde{R}_i \setminus (\partial \tilde{R}_i \cap \partial Q)$  be the boundary  $\partial \tilde{R}_i$  but not belong to  $\partial Q$ .

**Proposition 7.** Given  $p'_i \in H_i$ , there is a.e.  $C^1$  diffeomorphism  $\varphi_i$  between  $R_i^{\text{in}}$  and  $\tilde{R}_i^{\text{in}} \setminus \{p'_i\}$ , where  $R_i^{\text{in}}$  and  $\tilde{R}_i^{\text{in}}$  are the simplified representations of  $R_i^{\text{int}} \cup \partial R_i^{\text{int}}$  and  $\tilde{R}_i^{\text{int}} \cup \partial \tilde{R}_i^{\text{int}}$ , respectively.

*Proof.* The situation is categorized according to the number of the common edges of the subregion and the obstacle.

*Situation 1 (no common edge).* Since  $\partial H_i \cap \partial \tilde{R}_i = \emptyset$ , the diffeomorphism can be constructed using the method in the proof of Lemma 1 in [16].

*Situation 2 (one common edge in Figure 4(a)).* Draw a radial line  $l$  from  $p'_i$  to cross  $H_i$  and intersect with the boundaries of  $\tilde{R}_i$  and  $H_i$  at the points  $p_1$  and  $p_0$ , respectively. Note that if

the radial line lies on  $\partial \tilde{R}_i^{\text{in}}$ ,  $p_0$  and  $p_1$  are the vertices of  $H_i$  and  $\tilde{R}_i$  on the corresponding boundary, that is, points  $a_1(a_2)$  and  $b_1(b_2)$ . Obviously, it can be obtained that  $p_1 \neq p_0$ . Then a simple diffeomorphism  $\varphi_i$  can be constructed as

$$\begin{aligned} \varphi_i(q) &= \frac{\|p_1 - p'_i\|}{\|p_1 - p'_i\| - \|p_0 - p'_i\|} (\|q - p'_i\| - \|p_0 - p'_i\|). \end{aligned} \quad (23)$$

So any  $q \in R_i^{\text{in}}$  on the radial line  $l$  can be mapped to  $\varphi_i(q) \in \tilde{R}_i^{\text{in}} \setminus \{p'_i\}$ . Due to our hypothesis on  $\tilde{R}_i$  and  $R_i$ , it can be seen that  $\varphi_i$  is continuous and differentiable everywhere in  $R_i^{\text{in}}$ , and it has a unique inverse  $\varphi_i^{-1}$ , which is also continuous and differentiable everywhere in  $\tilde{R}_i^{\text{in}} \setminus \{p'_i\}$ . Therefore,  $\varphi_i$  is a  $C^1$  diffeomorphism. Note that the intersection points  $p_1$  and  $p_0$  are the functions of  $q$ . So even if  $q_1$  and  $q_2$  are equidistant to  $p'_i$ , we still have  $\varphi_i(q_1) \neq \varphi_i(q_2)$ . Hence, the inverse exists and  $\varphi_i$  is a diffeomorphism.

*Situation 3 (two common edges in Figure 4(b)).* The diffeomorphism is constructed as (23) in Situation 2.

*Situation 4 (three common edges).* There are two cases.

(i) For the first case in Figure 4(d), we first define the critical vertex as the common vertex of the subregion and

obstacle, while one of its adjacent vertices does not belong to the obstacle, such as  $p'_i$  and  $p_2$ . Choose one of the critical vertices as  $p'_i$ , then  $p''$  is on the same edge with the other critical vertex, and it does not belong to the obstacle. Thus the convex region is divided into two subregions by the blue dividing line connecting  $p'_i$  with  $p''$ . The first diffeomorphism  $\varphi_{ia}(q)$  can be constructed by drawing a radial line from  $p'_i$  using (23) as in Situation 2. For the other part, the diffeomorphism should be modified a little by drawing a radial line from  $p''$ , as follows:

$$\varphi_{ib}(q) = \frac{\|p_1 - p''\|}{\|p_0 - p''\|} (\|q - p''\|), \quad (24)$$

where  $p_1$  and  $p_0$  are the two intersection points of the radial line with the boundaries of the obstacle, and if  $q$  lies on the edge  $\partial\tilde{R}_i^{\text{in}}$ ,  $p_0$  and  $p_1$  are the corresponding vertices of the obstacle.

Note that, through the diffeomorphism  $\varphi_{ib}(q)$ , all the points (except  $p'_i$  and  $p''$ ) on the dividing line have the same mapping with the first diffeomorphism  $\varphi_{ia}(q)$ . And  $p''$  has a mapping by the diffeomorphism  $\varphi_{ia}$ . So we obtain  $\varphi_i = \varphi_{ia} \cup \varphi_{ib}$  that can map  $R_i^{\text{in}}$  to  $\tilde{R}_i^{\text{in}} \setminus \{p'_i\}$ .

(ii) For the second case in Figure 4(e), two edges of  $H_i$  are parallel which is a typical situation in structural environment. First, the convex polygon is divided into two partitions such that only one partition contains the obstacle. For the partition without any obstacle, the diffeomorphism is constructed as  $\varphi_{ia}(q) = q$ . For the other partition, a parallel line is drawn from  $q$  and intersects with the boundaries of  $\tilde{R}_i$  at the left point  $p_0$  and right point  $p_3$ . Given the vertices of the polygon,  $p_0$  lying on the line  $\overline{b_1c_1}$  can be mapped to  $p_1$  on the line  $\overline{p'_ic_1}$ , described as  $p_1 = p'_i - (\|p'_i - c_1\| \|b_1 - p_0\|) / (\|b_1 - c_1\|)$ . Draw another parallel line from  $p_1$ , and it intersects with the boundary of  $\tilde{R}_i$  at point  $p_2$ . Thus the diffeomorphism can be constructed as  $\varphi_{ib}(q) = p_1 + (\|q - p_0\| \|p_2 - p_1\|) / (\|p_3 - p_0\|)$ . Combining with the representation of  $p_1$ , the diffeomorphism can be described as

$$\varphi_{ib}(q) = p'_i - \frac{\|p'_i - c_1\| \|b_1 - p_0\|}{\|b_1 - c_1\|} + \frac{\|q - p_0\| \|p_2 - p_1\|}{\|p_3 - p_0\|}. \quad (25)$$

For this case,  $\varphi_i = \varphi_{ia} \cup \varphi_{ib}$ .

*Situation 5 (more than three common edges in Figure 4(e)).* The diffeomorphism is constructed as in Situation 4.

*Situation 6 (a special case).* All edges of the obstacle are common edges in Figure 4(f). The diffeomorphism is constructed as (24) by choosing the only vertex of the subregion which does not belong to obstacle as  $p''$ .

As claimed in the above six situations, it is obvious that the diffeomorphism  $\varphi_i$  can be constructed for each subregion.  $\square$

*Remark 8.* The subregion containing no obstacle will be transformed by an identical transformation, which is also a diffeomorphism.

*Remark 9.* Let  $\bigcup_{i=1}^M p'_i = p'$ . For each decomposed subregion, the diffeomorphism  $\varphi_i$  is constructed. So the transformation  $\varphi : F^{\text{int}} \rightarrow Q^{\text{int}} / \{p'\}$  is defined by  $\varphi(q) = \varphi_i(q)$ , where  $q \in R_i^{\text{in}}$ .

*Remark 10.* Given the search region and obstacles, the diffeomorphism can be computed off-line and stored in a table. Therefore, in the search procedure, the diffeomorphism can be directly used by looking up the table, which will significantly reduce the computation time.

*4.1.3. Numerical Examples.* Two examples of applying the convex decomposition method to divide the nonconvex region with arbitrary obstacles are shown in Figure 5. The green lines are the decomposing lines for convex decomposition of the concave obstacle, and the black lines are the dividing lines for the whole environment. In Figure 5(a), the concave obstacle is divided into three convex polygons. And the whole region is divided into four subregions, three of which containing one obstacle and one containing none. In Figure 5(b), a more complex nonconvex region with two obstacles is given. First, the whole region is divided into two partitions. Then the partition with a concave obstacle is divided into six subregions by using the decomposition method.

Further, the diffeomorphism is constructed for each example region by using the method in Proposition 7. We sample some points in the original regions and then map them to the transformed regions by the diffeomorphism, as shown in Figure 6.

*4.2. Cooperative Search Method in Nonconvex Environment with Arbitrary Obstacles.* The diffeomorphism that transforms a complex nonconvex region with arbitrary obstacles to a convex region has been constructed. Now we will propose the cooperative search method in the nonconvex environment based on the transformation.

*4.2.1. Objective Function Transformation.* The objective function (3) should be modified by shifting the allowable search domain from the feasible region  $F^{\text{int}}$  to the whole region  $Q^{\text{int}} \setminus \{p'\}$ . Based on the modification of Variables Theorem [29], it claims that if  $\Omega \subset \mathbb{R}^n$  is open,  $g_1 : \Omega \rightarrow \mathbb{R}$  is integrable over  $\Omega$ , and  $g_2 : \tilde{\Omega} \rightarrow \Omega$  is a  $C^1$  diffeomorphism, then  $(g_1 \circ g_2) |\det J_{g_2}|$  is integrable over  $\tilde{\Omega}$ , and

$$\int_{\Omega} g_1(\omega) d\omega = \int_{\tilde{\Omega}} (g_1 \circ g_2)(\tilde{\omega}) |\det J_{g_2}(\tilde{\omega})| d\tilde{\omega}, \quad (26)$$

where  $|\det J_{g_2}(\tilde{\omega})|$  is the determinant of the Jacobian of  $g_2$ . In this paper, we assume that  $g_1 = f(\|p_i - r\|)\gamma(r) : F^{\text{int}} \rightarrow \mathbb{R}$  and  $g_2 = \varphi^{-1} : Q^{\text{int}} / (p') \rightarrow R^{\text{int}}$ . Since  $\varphi^{-1}(\omega) = r$ ,

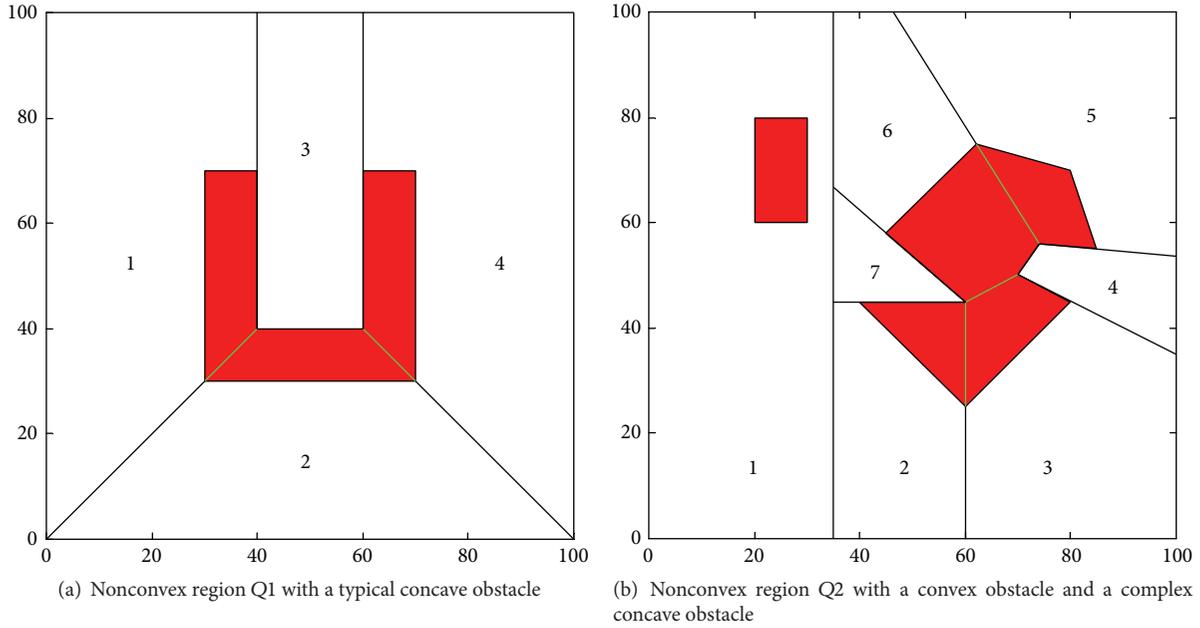


FIGURE 5: Convex decomposition.

the transformed objective function in the new convex region can be expressed as

$$\begin{aligned} & \int_{F^{\text{int}}} f(\|p_i - r\|) \gamma(r) dr \\ &= \int_{Q^{\text{int}} \setminus \{p'\}} f(\|\varphi^{-1}(q_i) - \varphi^{-1}(\omega)\|) \gamma(\varphi^{-1}(\omega)) \\ & \quad \cdot |\det J_{\varphi^{-1}}(\omega)| d\omega. \end{aligned} \quad (27)$$

It can be seen that the allowable search domain is transformed from the feasible region  $F^{\text{int}}$  to the whole region  $Q^{\text{int}} \setminus \{p'\}$ , while keeping the function value unchanged. However, the control law for the above objective function (the right side of (27)) is quite hard to be obtained. In order to use the Lloyd algorithm, the new objective function is defined as

$$H'(P) = \int_{F^{\text{int}}} f(\|\varphi(p_i) - \varphi(r)\|) \gamma(r) dr. \quad (28)$$

Since  $\varphi$  is the diffeomorphism,  $F^{\text{int}}$  and  $\varphi(F^{\text{int}})$  are topologically equivalent, and the problem remains essentially unchanged. Further, the corresponding function in the transformed domain is obtained:

$$\begin{aligned} & H(W, \bar{V}) \\ &= \int_{Q^{\text{int}} \setminus \{p'\}} f(\|q_i - \omega\|) \gamma(\varphi^{-1}(\omega)) |\det J_{\varphi^{-1}}(\omega)| d\omega. \end{aligned} \quad (29)$$

So the gradient descent approach as (6) can be used to minimize the objective function

$$\dot{q}_i = -k_i^1 (q_i - \bar{C}_{V_i}), \quad (30)$$

where  $k_i^2$  is a positive gain,  $q_i = \varphi(p_i)$  is the transformed position of  $p_i$  by the diffeomorphism in the transformed region  $Q^{\text{int}} \setminus \{p'\}$ , and  $\bar{C}_{V_i}$  is the centroid of the Voronoi region generated by  $q_i$ .

**4.2.2. Control Strategy for Cooperative Search in the Nonconvex Environment.** Since the diffeomorphism  $\varphi$  is a bijection, the trajectory of  $q_i$  in  $Q^{\text{int}} \setminus \{p'\}$  has the unique preimage in original region  $F^{\text{int}}$ , which is the trajectory of  $p_i$ . So the control law for the new objective function (28) in the original region can be obtained by the inverse mapping of (30):

$$\dot{p}_i = -k_i^2 J_{\varphi}^{-1}(\varphi(p_i)) (p_i - \varphi^{-1}(\bar{C}_{V_i})), \quad (31)$$

where  $k_i^2$  is a positive gain and  $J_{\varphi}$  is the Jacobian of  $\varphi$ .

The following is an analysis of the convergence of the control law (31) using LaSalle's invariance principle [30], which is basically an extension of Lyapunov's theorem [31].

**Theorem 11.** *In the nonconvex region, the trajectories of UAVs governed by the control law (31), starting from any initial condition  $P(0) = (p_{1,0}, \dots, p_{N,0}) \in F^{\text{int}}$ , will asymptotically converge to the desired location.*

*Proof.* Since the diffeomorphism  $\varphi$  is a bijection, the trajectory  $q_i$  in  $Q^{\text{int}} \setminus \{p'\}$  has the unique preimage  $p_i$  in original region  $F^{\text{int}}$ . Therefore, to prove that the trajectory  $p_i$  governed by the control law (31), starting from any initial condition  $P(0)$ , can converge to  $\varphi^{-1}(\bar{C}_{V_i})$ , we only need to prove that the trajectory  $q_i$  governed by the control law (30), starting from initial condition  $W(0) = \varphi(P(0)) = (\varphi(p_{1,0}), \dots, \varphi(p_{N,0}))$ , can converge to  $\bar{C}_{V_i}$ .

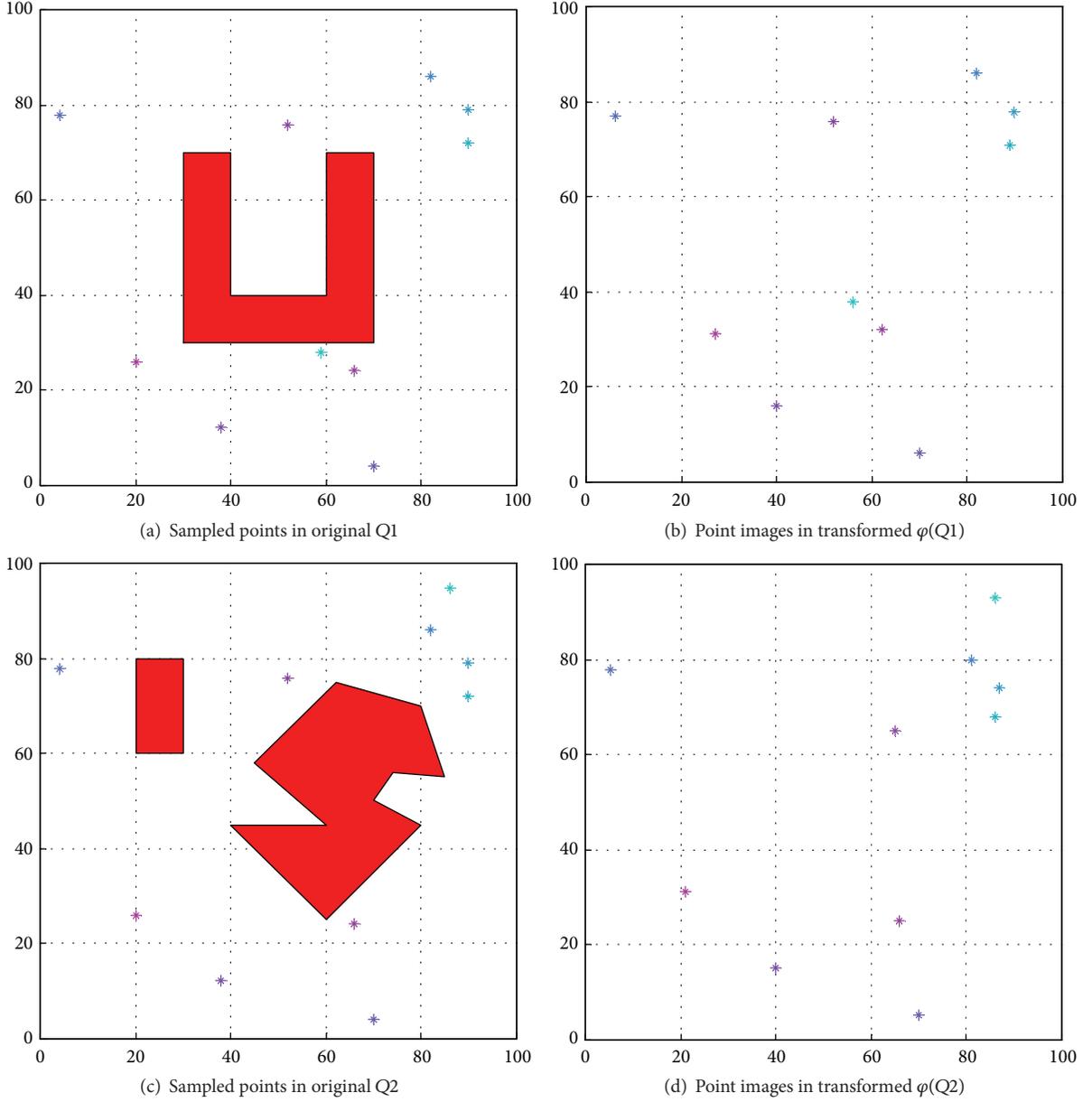


FIGURE 6: Diffeomorphism of the nonconvex region. (a) and (c) are the two examples of the original nonconvex region with 10 randomly sampled points. (b) and (d) are the transformed convex regions with the transformed image of the sampled points by diffeomorphism.

Assume  $V(Q) = H(W, \tilde{V}) = \sum_{i=1}^N \int_{\tilde{V}_i} \|q_i - \omega\|^2 \gamma(\varphi^{-1}(\omega)) |\det J_{\varphi^{-1}}(\omega)| d\omega$ , where  $Q = (q_1, q_2, \dots, q_N)$  represent the transformed positions of UAVs in  $Q^{\text{int}} \setminus \{p'\}$  through the diffeomorphism  $\varphi$ . Consider

$$\begin{aligned} \dot{V}(Q) &= \frac{dH(W, \tilde{V})}{dt} = \sum_{i=1}^N \frac{\partial H(W, \tilde{V})}{\partial q_i} \dot{q}_i \\ &= \sum_{i=1}^N 2\lambda \tilde{M}_{V_i} (q_i - \tilde{C}_{V_i}) (-k_i^{-1} (q_i - \tilde{C}_{V_i})) \end{aligned}$$

$$= -2\lambda_1 k_i^{-1} \sum_{i=1}^N \tilde{M}_{V_i} (q_i - \tilde{C}_{V_i})^2 \leq 0, \quad (32)$$

where  $\lambda_1 = |\det J_{\varphi^{-1}}(\omega)|$  and  $\tilde{M}_{V_i}$  is the mass of the Voronoi region generated by  $q_i$  in the convex region  $Q^{\text{int}} \setminus \{p'\}$ .

The following can be observed:

- (i)  $V: Q^{\text{int}} \setminus \{p'\} \rightarrow \mathbb{R}$  is continuously differentiable in  $Q^{\text{int}} \setminus \{p'\}$ .
- (ii)  $Q^{\text{int}} \setminus \{p'\}$  is a compact invariant set.
- (iii)  $\dot{V}$  is negative semidefinite in  $Q^{\text{int}} \setminus \{p'\}$ .

(iv)  $E = \dot{V}^{-1}(0) = \{\bar{C}_{V_i}\}$ .

(v)  $E$  itself is the largest invariant subset of  $E$  by the control law (30).

Thus, by LaSalle's invariance principle, the trajectories  $q_i$  governed by (30), starting from initial configuration  $W(0)$ , will asymptotically converge to the critical points of  $H(W, \bar{V})$ ; that is, the trajectories of UAVs governed by the control law (31), starting from any initial condition  $P(0) = (p_{1,0}, \dots, p_{N,0}) \in F^{\text{int}}$ , will asymptotically converge to the desired location.  $\square$

Therefore, the discrete-time motion model and the real control law for UAV  $i$  in the original nonconvex region can be written as

$$\begin{aligned} p_{i,m} &= p_{i,m-1} + u_{i,m}, \\ u_{i,m} &= -k_i^2 J_\varphi^{-1}(\varphi(p_{i,m-1}))(p_{i,m-1} - \varphi^{-1}(\bar{C}_{V_i})). \end{aligned} \quad (33)$$

Further, considering the mobility of the UAV, the control law must meet some constraints such that the generated trajectories are kinematically feasible:

(1) Maximum velocity: the maximum velocity  $v_{\max}$  will limit the control law by the following saturation rule:

$$u_{i,m} = \begin{cases} u_{i,m}, & \|u_{i,m}\| \leq v_{\max} \\ \frac{v_{\max}}{\|u_{i,m}\|} u_{i,m}, & \text{Otherwise.} \end{cases} \quad (34)$$

where  $u_{i,m} = -k_i^2 J_\varphi^{-1}(\varphi(p_{i,m-1}))(p_{i,m-1} - \varphi^{-1}(\bar{C}_{V_i}))$ , as shown in (33).

**4.2.3. Cooperative Search Algorithm in the Nonconvex Environment with Arbitrary Obstacles.** The cooperative search algorithm for the nonconvex region with arbitrary obstacles is presented as Algorithm 1.

*Algorithm 1.* Cooperative search algorithm in the nonconvex environment with arbitrary obstacles is as follows:

- (1) Initially, all the UAVs are randomly deployed in the feasible region. The probability map and uncertainty map of each UAV are initialized.
- (2) The nonconvex region with arbitrary obstacles is divided by the global decomposition method, and

(2) Maximum turning angle: the maximum turning angle during a unit time period  $\Delta t$  is derived from the turning rate  $\omega_{\max}$ , described as  $\theta_{\max} = \omega_{\max} \Delta t$ . Thus the control law is limited by the following equations:

$$u_{i,m} = \begin{cases} u_{i,m}, & \arccos \frac{u_{i,m} u_{i,m-1}}{\|u_{i,m} u_{i,m-1}\|} \leq \theta_{\max} \\ \frac{\|u_{i,m}\|}{\|u_{i,m-1}\|} u_{i,m-1} A, & \text{Otherwise,} \end{cases} \quad (35)$$

where  $A$  is the rotating matrix for the maximum turning angle from the last control input as

$$A = \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix}, \quad (36)$$

$$\alpha = \begin{cases} \theta_{\max}, & u_{i,m} \cdot u_{i,m-1} \geq 0 \\ -\theta_{\max}, & u_{i,m} \cdot u_{i,m-1} < 0 \end{cases}$$

represents the rotating direction.

Combining (34) with (35), the control law can be expressed as

$$u_{i,m} = \begin{cases} u_{i,m}, & \|u_{i,m}\| \leq v_{\max} \text{ and } \arccos \frac{u_{i,m} u_{i,m-1}}{\|u_{i,m} u_{i,m-1}\|} \leq \theta_{\max} \\ \frac{\|u_{i,m}\|}{\|u_{i,m-1}\|} u_{i,m-1} A, & \|u_{i,m}\| \leq v_{\max} \text{ and } \arccos \frac{u_{i,m} u_{i,m-1}}{\|u_{i,m} u_{i,m-1}\|} > \theta_{\max} \\ \frac{v_{\max}}{\|u_{i,m}\|} u_{i,m}, & \|u_{i,m}\| > v_{\max} \text{ and } \arccos \frac{u_{i,m} u_{i,m-1}}{\|u_{i,m} u_{i,m-1}\|} \leq \theta_{\max} \\ \frac{v_{\max}}{\|u_{i,m-1}\|} u_{i,m-1} A, & \text{Otherwise,} \end{cases} \quad (37)$$

the diffeomorphism  $\varphi$  is constructed as described in Proposition 7.

- (3) Through the diffeomorphism  $\varphi$ , the original nonconvex region  $F^{\text{int}}$  is transformed into the new convex one,  $Q^{\text{int}} \setminus \{p'\}$ . And the positions of UAVs are also transformed into new virtual ones.
- (4) The Voronoi partition of transformed region  $Q^{\text{int}} \setminus \{p'\}$  is generated by the virtual positions of UAVs. Compute the centroid  $\bar{C}_{V_i}$  of each Voronoi region using probability map as the density.
- (5) At each time step, UAV  $i$  moves towards the preimages of its generalized Voronoi centroid  $\varphi^{-1}(\bar{C}_{V_i})$  in the original region, governed by the control law (31).

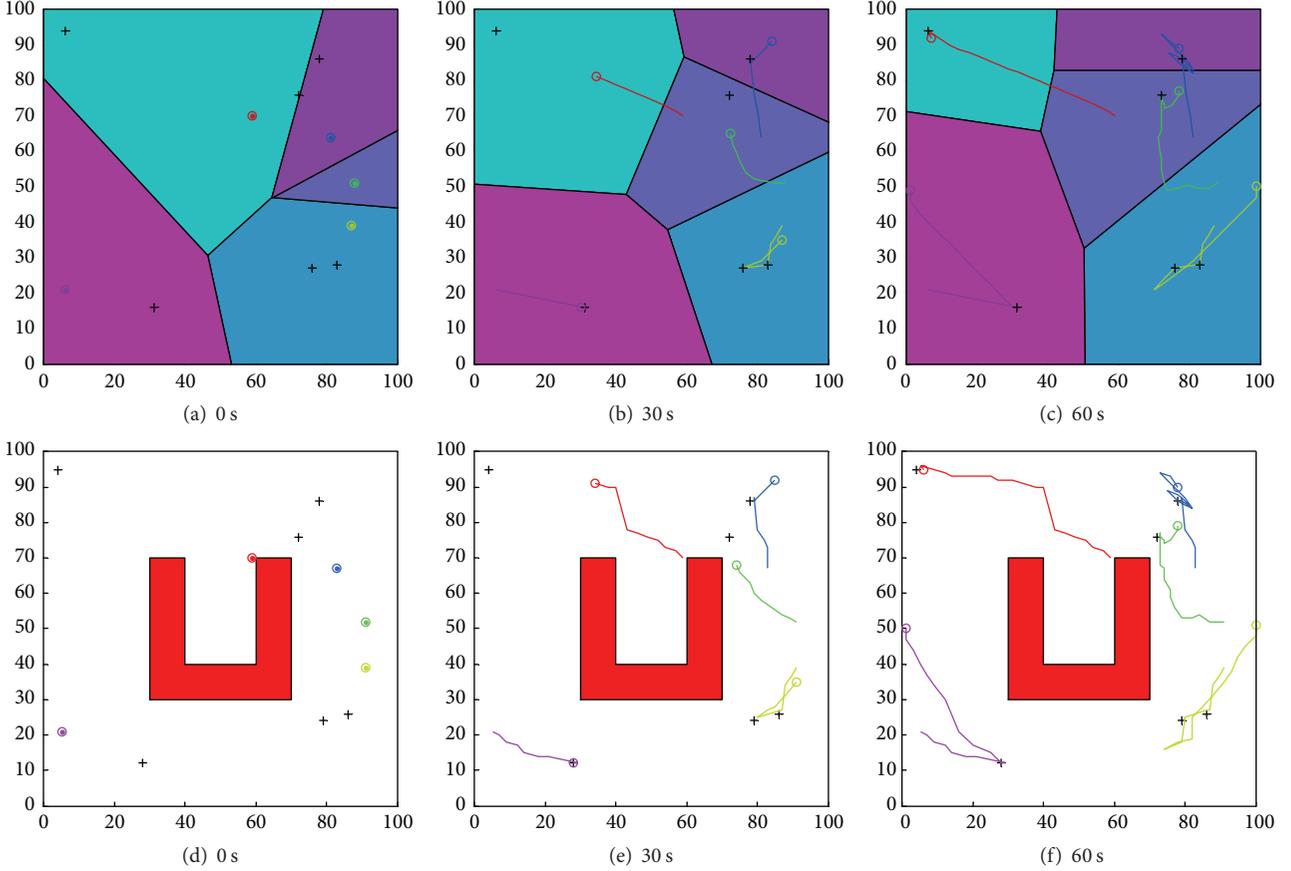


FIGURE 7: Search flight trajectories of UAVs with time: (a–c) in the transformed region and (d–f) in the original region.

- (6) Update the probability map  $TP_i$  of UAV  $i$ . If the target probability of a cell is greater than  $P_{\max}$ , a target is found.
- (7) Update the Voronoi partitions based on the new virtual positions of UAVs. Compute the generalized Voronoi centroids  $\tilde{C}_{V_i}$ .
- (8) If UAV  $i$  reaches the preimage of its Voronoi centroid  $\varphi^{-1}(\tilde{C}_{V_i})$  and does not move, then use its uncertainty map instead of probability map to compute the new Voronoi centroid  $\varphi^{-1}(\tilde{C}_{V_i})$  from this time step.
- (9) If the terminal condition is satisfied, the algorithm ends; otherwise, go to (5).

## 5. Simulation

In this section, some experiments are performed to verify the effectiveness of the cooperative search algorithm in different scenarios, and then the comparison with other algorithms is conducted to demonstrate the advantage of our algorithm in addressing the cooperative search problem in a nonconvex environment.

**5.1. Simulation Environment.** The algorithms have been programmed in Matlab. The simulation scenario is set as that

five UAVs search for targets in two typical nonconvex regions with differently shaped obstacles. The whole search region  $Q$  is defined as a square region of  $[0, 10] \times [0, 10]$  km<sup>2</sup>. And the distribution of each target follows the Gaussian distribution function

$$\phi(r) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(r-\mu)^2}{2\sigma^2}}, \quad (38)$$

where the mean value  $\mu$  is the expected position of the target and  $\sigma$  is set to be 1 km, representing the predefined probability distribution range. The communication range  $R_C$  is 5 km. Initially, all the UAVs are randomly deployed in the feasible region. Some other parameters for simulation are in Table 1.

### 5.2. Simulation Results and Analysis

**5.2.1. Scenario 1: Nonconvex Environment with Concave Obstacles.** The first obstacle area is a typical concave region, represented as a red square. The convex decomposition and the diffeomorphism have been implemented as shown in Figures 5(a), 6(a), and 6(b). Then the cooperative search algorithm is performed. The trajectories of UAVs at different time steps in the transformed region without obstacles and in the original region are shown in Figure 7, where the UAVs and targets are denoted by “o” and “+”, respectively.

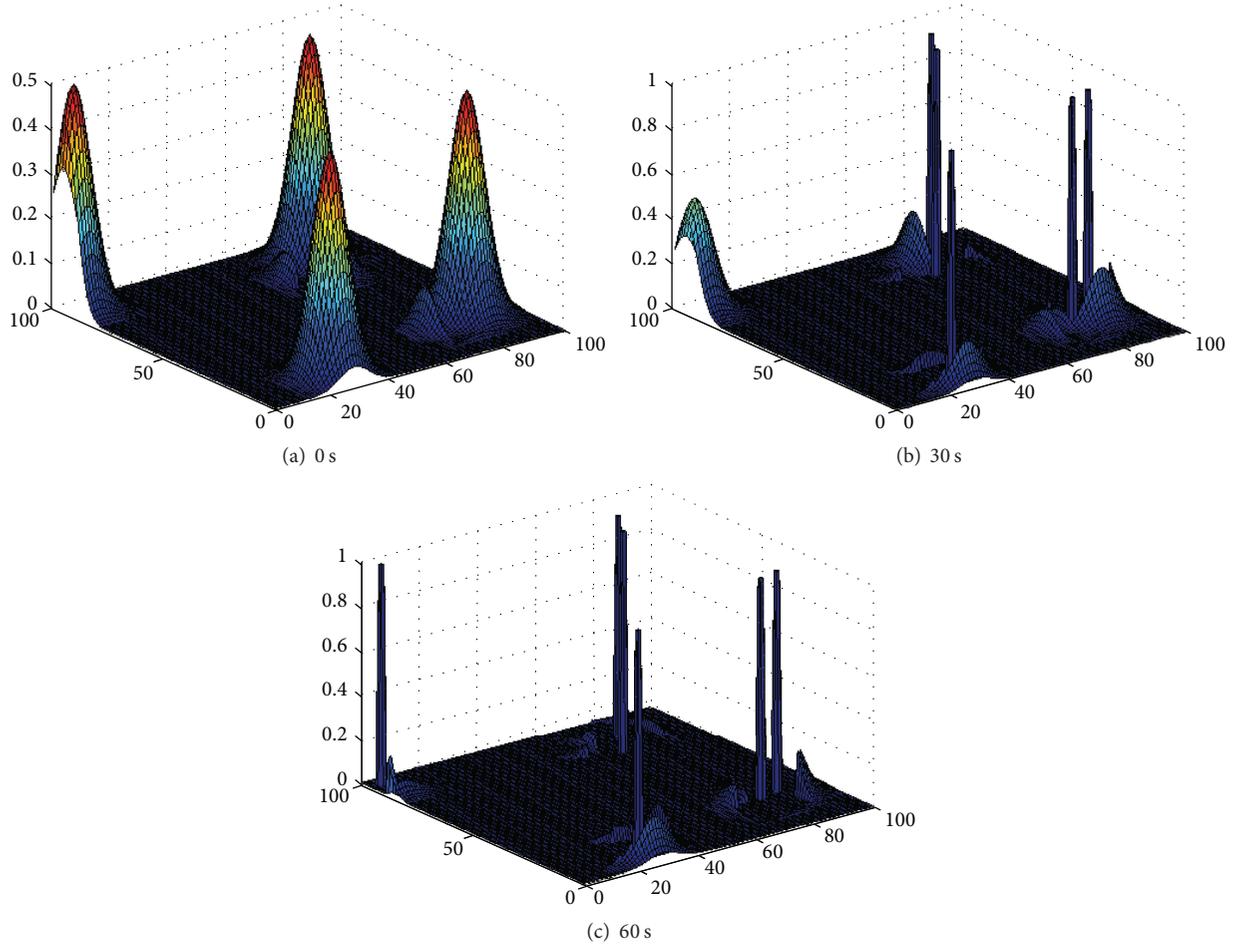


FIGURE 8: Probability map update for Scenario 1.

TABLE 1: Simulation parameters.

UAV	Sensor			Detection model	
Maximum speed (m/s)	100	$\alpha_s$	45°	Positive sensing probability $P_{0i}$	0.9
Maximum turning angle	60°	$\beta_s$	30°	False alarm probability $p_a$	0.1
Flight height (m)	100	$\gamma_s$	30°	Target existence criteria $P_{max}$	0.9

The probability map with time evolving is shown in Figure 8. From the figures, it can be seen that the probability map has converged. UAVs in the original region, starting from any initial positions, can find all the 6 targets at time 45 s while avoiding the obstacle.

**5.2.2. Scenario 2: Nonconvex Environment with Several Complex Obstacles.** The nonconvex region in Scenario 2 contains two obstacles. The results of cooperative search are shown in Figures 9 and 10. It can be seen that the trajectories of UAVs in the original region governed by the control law (31), starting from any initial positions, can move towards the targets while avoiding the obstacle. It also can be seen that the probability map has converged. After finding the targets, the UAV uses the uncertainty map instead of the probability map to escape from the local optimum (the positions of targets with high

target existence probability) and keep searching. From the simulation results, all the targets are found at time 54 s.

**5.2.3. Comparisons with Other Algorithms.** In order to verify the performance, the proposed algorithm is compared with a sweep-line search algorithm [7] and a random search algorithm [32]. In the sweep-line search algorithm, the search mood is fixed, and each UAV sweeps the region parallel to the boundary and decides the width between two search lines based on the detection range. In the random search algorithm, each UAV randomly chooses a cell as the next desired flight point without considering the target information. If the choice conflicts with other UAVs or the next flight point conflicts with the obstacle, choose another one.

We use the same simulation environment in Scenario 1. The above three algorithms are implemented and compared

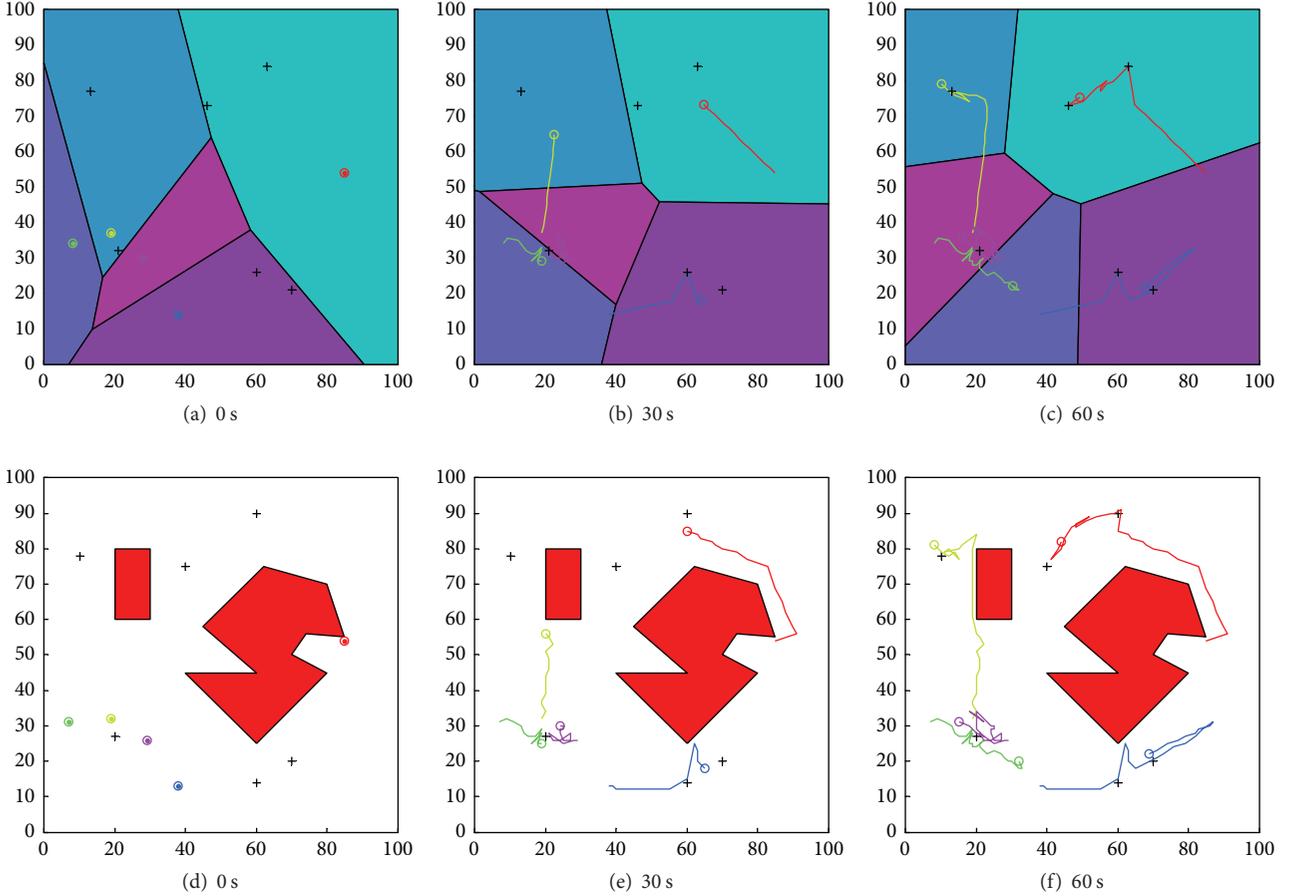


FIGURE 9: Search flight trajectories of UAVs with time: (a–c) in the transformed region and (d–f) in the original region.

TABLE 2: Results of different search algorithms.

	Sweep-line search algorithm	Random search algorithm	Our cooperative search algorithm
Average time required to search out all the targets (s)	461	N/A	56
Success rate in 500 s	64%	2%	94%
Repeated detection rate in 500 s	4.64	17.74%	6.45%

from the following aspects: average time required to find all the 6 targets, success rate of finding all the targets in a given time period of 500 s, and repeated detection rate. All the data are averaged by 50 runs. The results are shown in Table 2.

From Table 2, it can be seen that the average time for our proposed cooperative search algorithm to find all the targets is 56 s, which is more efficient and effective than both the sweep-line (461 s) and the random search algorithm (N/A, too many failures). For the second item, the success rate in 500 s of the cooperative search algorithm is 94%, which is much larger than that of the sweep-line and random search algorithm. The success rate of the sweep-line search algorithm is 64% due to the limited task time of 500 s,

while the random search algorithm has an unsatisfying success rate of 2%. For the third item, our proposed cooperative search algorithm performs the repeated detection rate as 6.45% when escaping from the local optimum to find new targets. And the repeated search of the sweep-line algorithm only occurs when turning as 4.64%. However, the random search algorithm may search for the same cell frequently with a repeated detection rate of 17.74%.

Further, 20 targets are randomly deployed in the nonconvex region. The number of targets found with time evolving is used as the measure of performance, as shown in Figure 11. According to the simulation results, the proposed cooperative search algorithm can lead UAVs to find more targets in the same time duration than others, because the control law leads

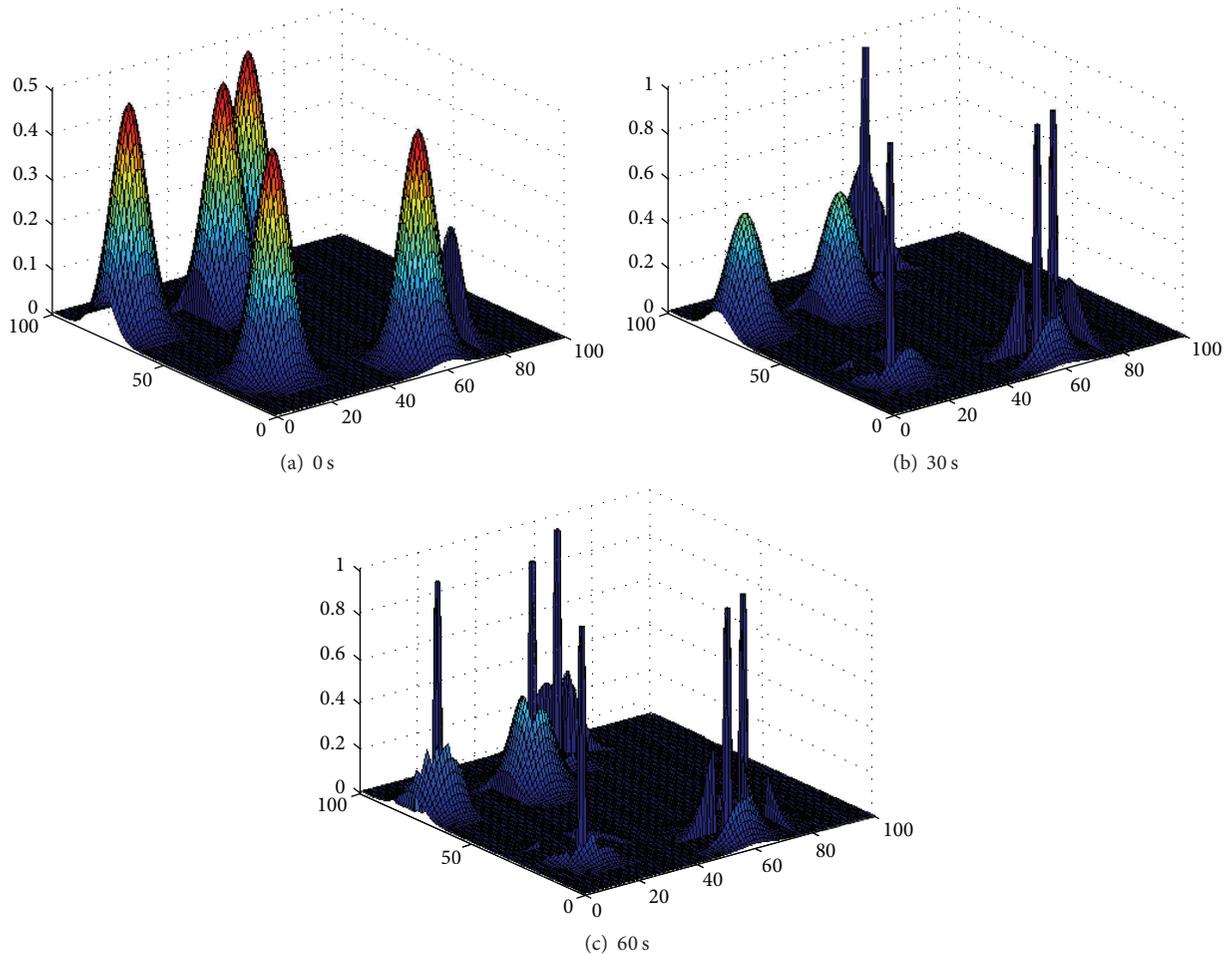


FIGURE 10: Probability map update for Scenario 2.

UAV to converge to the regions with high target existence probability. And if time is sufficient, the sweep-line algorithm will also find all the targets. However, the random search performs poorly due to its random mode.

From all the results, it can be seen that our proposed cooperative search algorithm is quite efficient, for the search is led by the control law to optimize the information related objective function. And it can not only lead UAVs to converge to the regions with high target existence probability but also improve the exploration of the uncertain regions. Besides, the nonconvex environment is transformed into a convex one through the diffeomorphism at the beginning, so there is no excess step to deal with obstacle avoidance during the search process, dramatically speeding up the convergence.

## 6. Conclusion and Future Work

This paper presents a distributed cooperative search method with multiple UAVs in a complex nonconvex environment with arbitrary obstacles. The proposed method mainly focuses on two issues: how to integrate the information update into the coverage control scheme and how to address the search problem in a complex nonconvex environment.

First, an asynchronous distributed cooperative search framework is proposed, integrating information update into the coverage control scheme. The information update includes the Bayesian theory based individual update and a consensus fusion protocol based fusion update, while taking the inaccuracies and uncertainties in the sensor information into consideration. And an adaptive density function of coverage optimization is formulated based on the probability and uncertainty maps. Second, a new transformation method is proposed in order to extend the diffeomorphism to deal with the nonconvex environment with concave obstacles. The control strategy is presented considering the kinematic constraints, and the convergence is proved by LaSalle's invariance principle. Finally, the effectiveness of the proposed algorithm is demonstrated through simulations. In future work, we will extend the algorithm to a high-dimension space and take the complex dynamics of the UAV into consideration.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

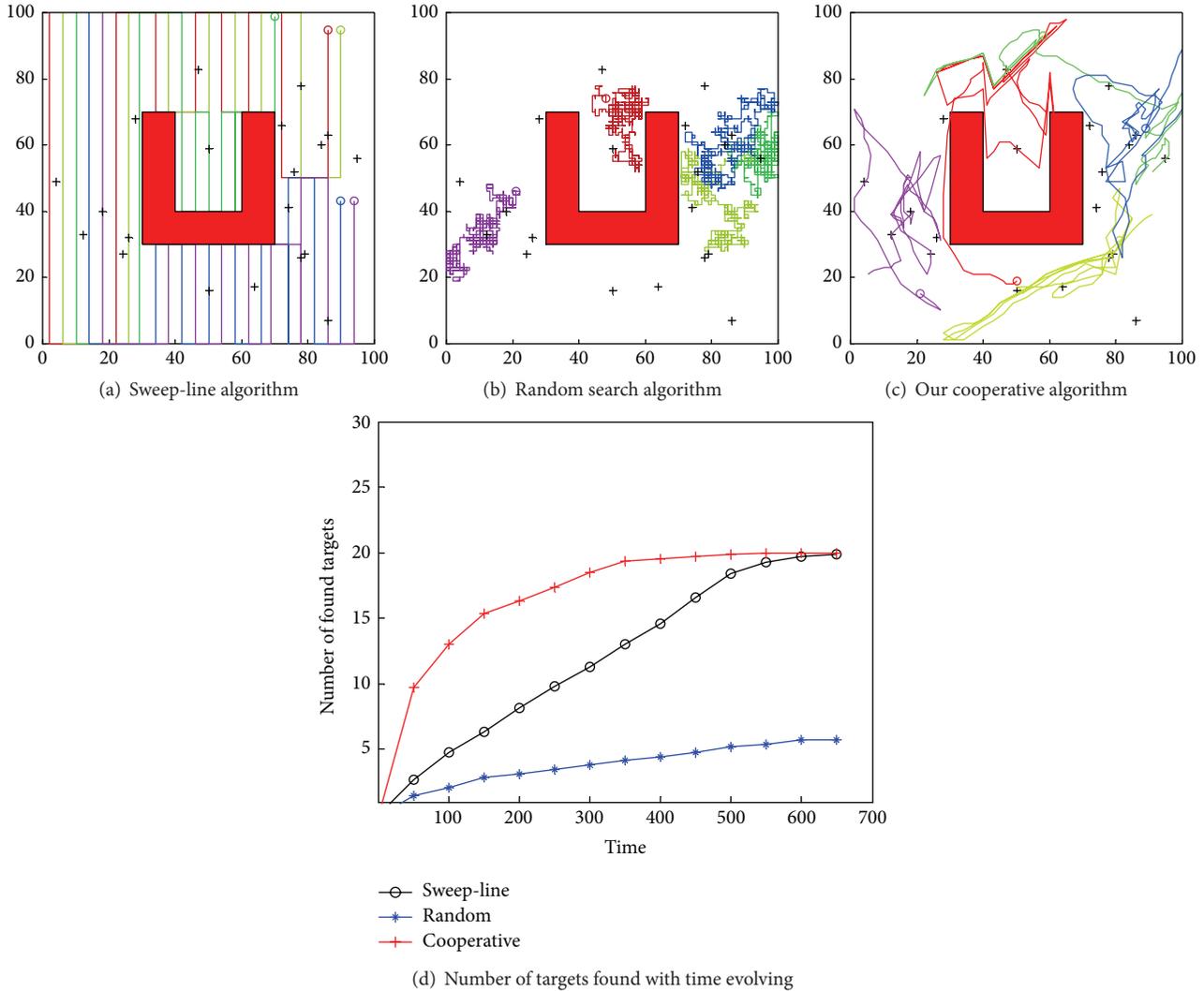


FIGURE II: Comparison with sweep-line search and random search algorithm.

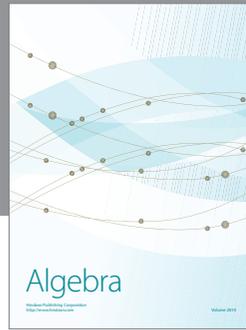
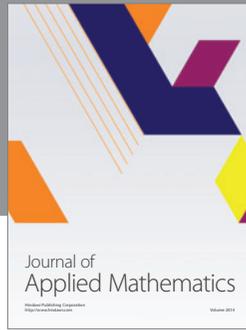
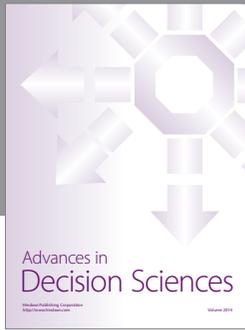
## Acknowledgments

This study was supported by the grant from the National Natural Science Foundation of China (61403406 and 61403410) and supported by the research project of National University of Defense Technology.

## References

- [1] E. P. de Freitas, T. Heimfarth, A. M. Ferreira, C. E. Pereira, F. R. Wagner, and T. Larsson, "Decentralized task distribution among cooperative UAVs in surveillance systems applications," in *Proceedings of the 7th International Conference on Wireless On-demand Network Systems and Services (WONS '10)*, pp. 121–128, February 2010.
- [2] E. Kuiper and S. Nadjm-Tehrani, "Mobility models for UAV group reconnaissance applications," in *Proceedings of the 2nd International Conference on Wireless and Mobile Communications (ICWMC '06)*, pp. 33–43, July 2006.
- [3] J. Tisdale, A. Ryan, Z. Kim, D. Törnqvist, and J. K. Hedrick, "A multiple UAV system for vision-based search and localization," in *Proceedings of the American Control Conference*, pp. 1985–1990, IEEE, Seattle, Wash, USA, June 2008.
- [4] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas, "Cooperative air and ground surveillance," *IEEE Robotics and Automation Magazine*, vol. 13, no. 3, pp. 16–26, 2006.
- [5] Y. Yang, M. M. Polycarpou, and A. A. Minai, "Multi-UAV cooperative search using an opportunistic learning method," *Transactions of the ASME—Journal of Dynamic Systems, Measurement and Control*, vol. 129, no. 5, pp. 716–728, 2007.
- [6] J. Berger, J. Happe, C. Gagné, and M. Lau, "Co-evolutionary information gathering for a cooperative unmanned aerial vehicle team," in *Proceedings of the 12th International Conference on Information Fusion (FUSION '09)*, pp. 347–354, IEEE, Seattle, Wash, USA, July 2009.
- [7] W. H. Huang, "Optimal line-sweep-based decompositions for coverage algorithms," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 27–32, May 2001.
- [8] P. B. Sujit and D. Ghose, "Search using multiple UAVS with flight time constraints," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, no. 2, pp. 491–509, 2004.

- [9] I. I. Hussein and D. M. Stipanović, "Effective coverage control for mobile sensor networks with guaranteed collision avoidance," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 4, pp. 642–657, 2007.
- [10] J. W. Lee, B. Walker, and K. Cohen, "Path planning of unmanned aerial vehicles in a dynamic environment," in *Proceedings of the INFOTECH@AEROSPACE*, St. Louis, Mo, USA, 2011.
- [11] J. Cortés, S. Martínez, T. Karataş, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.
- [12] L. C. A. Pimenta, V. Kumar, R. C. Mesquita, and G. A. S. Pereira, "Sensing and coverage for a network of heterogeneous robots," in *Proceedings of the IEEE Conference on Decision and Control*, pp. 3947–3952, Cancun, Mexico, December 2008.
- [13] M. Zhong and C. G. Cassandras, "Distributed coverage control and data collection with mobile sensor networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 10, pp. 2445–2455, 2011.
- [14] M. Mirzaei M, F. Sharifi, B. W. Gordon, C. A. Rabbath, and Y. M. Zhang, "Cooperative multi-vehicle search and coverage problem in uncertain environments," in *Proceedings of the IEEE Conference on Decision and Control and European Control Conference*, pp. 4140–4145, 2012.
- [15] J. Hu, L. Xie, K.-Y. Lum, and J. Xu, "Multiagent information fusion and cooperative control in target search," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 4, pp. 1223–1235, 2013.
- [16] C. H. Caicedo-Núñez and M. Žefran, "Performing coverage on nonconvex domains," in *Proceedings of the 17th IEEE International Conference on Control Applications (CCA '08)*, pp. 1019–1024, September 2008.
- [17] C. H. Caicedo-Núñez and M. Žefran, "A coverage algorithm for a class of non-convex regions," in *Proceedings of the 47th IEEE Conference on Decision and Control (CDC '08)*, pp. 4244–4249, December 2008.
- [18] A. Breitenmoser, M. Schwager, J.-C. Metzger, R. Siegwart, and D. Rus, "Voronoi coverage of non-convex environments with a group of networked robots," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '10)*, pp. 4982–4989, IEEE, Anchorage, Alaska, USA, May 2010.
- [19] S. Teraoka, T. Ushio, and T. Kanazawa, "Voronoi coverage control with time-driven communication for mobile sensing networks with obstacles," in *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, pp. 1980–1985, Orlando, Fla, USA, December 2011.
- [20] M. Adibi, H. A. Talebi, and K. Y. Nikravesh, "Adaptive coverage control in non-convex environments with unknown obstacles," in *Proceedings of the 21st Iranian Conference on Electrical Engineering (ICEE '13)*, pp. 1–6, May 2013.
- [21] J. W. Durham, R. Carli, P. Frasca, and F. Bullo, "Discrete partitioning and coverage control for gossiping robots," *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 364–378, 2012.
- [22] A. Ganguli, J. Cortés, and F. Bullo, "Multirobot rendezvous with visibility sensors in nonconvex environments," *IEEE Transactions on Robotics*, vol. 25, no. 2, pp. 340–352, 2009.
- [23] P. DeLima and D. J. Pack, "Maximizing search coverage using future path projection for cooperative multiple UAVs with limited communication ranges," in *Optimization and Cooperative Control Strategies*, vol. 381 of *Lecture Notes in Control and Information Sciences*, pp. 103–117, Springer, Berlin, Germany, 2009.
- [24] Z. Gu, *The Concept Introduction of Geometry and Topology*, High Education Press, Beijing, China, 2011.
- [25] B. Schachter, "Decomposition of polygons into convex sets," *IEEE Transactions on Computers*, vol. 27, no. 11, pp. 1078–1082, 1978.
- [26] B. Chazelle and D. P. Dobkin, "Optimal convex decompositions," *Computational Geometry*, vol. 4, no. 5, pp. 63–133, 1985.
- [27] J. M. Keil, "Decomposing a polygon into simple components," *SIAM Journal on Computing*, vol. 14, no. 4, pp. 799–817, 1985.
- [28] Y. Zhang, "An improved weighting algorithm for decomposing a simple polygon into convex polygons," *Journal of Yanshan University*, vol. 25, no. 1, pp. 76–79, 2001.
- [29] K. Janich, *Vector Analysis*, Springer, 2001.
- [30] H. J. Marquez, *Nonlinear Control Systems: Analysis and Design*, John Wiley & Sons, 2003.
- [31] J. P. LaSalle, "Some extensions of Liapunov's second method," *IRE Transactions on Circuit Theory*, vol. 7, no. 4, pp. 520–527, 1960.
- [32] Y. Yang, A. Minai, and M. M. Polycarpou, "Decentralized cooperative search by networked UAVs in an uncertain environment," in *Proceedings of the American Control Conference*, pp. 5558–5563, 2004.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

