Dynamic Discrete Pigeon-Inspired Optimization for Multi-UAV Cooperative Search-Attack Mission Planning

HAIBIN DUAN, Senior Member, IEEE
JIANXIA ZHAO
YIMIN DENG
Beihang University Beijing, China

YUHUI SHI, Fellow, IEEE
Southern University of Science and Technology Shenzhen, China

XILUN DING
Beihang University Beijing, China

For multiple unmanned aerial vehicles (UAVs) performing aerial search-attack tasks, there is a tradeoff between maximizing total benefit and minimizing consumption under the validity of constraints. This article proposes a dynamic discrete pigeon-inspired optimization algorithm to handle cooperative search-attack mission planning for UAVs, which integrates the centralized task assignment and distributed path generation aspects of the problem. Besides, a solution acceptance strategy is proposed to avoid frequent task switching. To design a reasonable objective function, the probability map is constructed and updated by Bayes formula to guide the following search motion, and a response threshold sigmoid model is adopted for target allocation during executing attack. Moreover, the flyable trajectories are generated by B-spline curves based on the simplified waypoints. Finally, numerical experiments prove that the proposed methods can provide feasible solutions for multiple UAVs considering different scenarios, such as the absence or presence of threats and insufficient resources. The results also show that the solution acceptance strategy is effective to improve performance. Moreover, the extensible mission planning system also integrates with an interactive 3D visualization simulation module, where the multi-UAV coordinated flight processes are demonstrated dynamically.

Manuscript received January 15, 2020; revised June 4, 2020 and August 7, 2020; released for publication September 27, 2020. Date of publication October 8, 2020; date of current version February 9, 2021.

DOI. No. 10.1109/TAES.2020.3029624

Refereeing of this contribution was handled by G. Inalhan.

This work was supported in part by Science and Technology Innovation 2030-Key Project of "New Generation Artificial Intelligence" under Grant #2018AAA0102303, in part by National Natural Science Foundation of China under Grant 91948204, Grant 91648205, Grant 61761136008, Grant U1913602, and Grant U19B2033, and in part by Aeronautical Foundation of China under Grant 20185851022.

Authors' addresses: Haibin Duan is with the State Key Laboratory of Virtual Reality Technology on Systems, School of Automation Science and Electrical Engineering, Beihang University, Beijing 100083, China, and also with Peng Cheng Laboratory, Shenzhen 518000, China, E-mail: (hbduan@buaa.edu.cn); Jianxia Zhao, and Yimin Deng are with the State Key Laboratory of Virtual Reality Technology on Systems, School of Automation Science and Electrical Engineering, Beihang University, Beijing 100083, China, E-mail: (zhaojianxia@buaa.edu.cn; ymdeng@buaa.edu.cn); Yuhui Shi is with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China, E-mail: (shiyh@sustc.edu.cn); Xilun Ding is with the School of Mechanical Engineering and Automation, Beihang University, Beijing 100083, China, E-mail: (xlding@buaa.edu.cn). *(Corresponding author: Haibin Duan)*.

I. INTRODUCTION

Nowadays, the rapid development of the unmanned aerial vehicle (UAV) [1] has promoted their incorporation into many fields to perform complicated tasks located in hazardous and even hostile environment [2]–[5]. Thus, UAVs are utilized for missions with low cost and excellent maneuverability while avoiding casualties. In recent years, the UAVs have attracted much attention in cooperative mission planning problems [6]–[9] due to the improvement of autonomy. Obviously, higher efficiency can be achieved with multiple UAVs operating in a coordinated manner [10]–[13], while the risks remarkably decrease in uncertain environments through active information exchange between the UAVs.

The Mission planning problem is a complicated optimization problem with various constraints, in consideration of task priority and coordination and trajectory feasibilities [6]. Many methods have been proposed to investigate the multi-UAV cooperative mission planning problem. Classic planners use genetic algorithm [13], [14], market-based method [15], [16], etc. Cooperative path planning and task assignment are mainly considered for multiple UAVs. For path planning, many mathematical and heuristic approaches have been presented to obtain the optimal route. Pitre *et al*. [17] proposed an information-value approach to gain more useful information for search-track missions by a single scalar index. Angley *et al*. [18] presented a multiobjective optimization model to optimize the search and survival strategies to localize a large number of target points for the UAVs. For task assignment, Chen *et al*. [19] modified the two-part wolf pack search algorithm which was applied to the time-sensitive multi-UAV cooperative task allocation problem. Zhou *et al*. [20] transformed the optimization problem into a two-sided two-stage optimal matching problem, in which the task assignment problem was addressed by a market mechanism, namely the Gale–Shapley algorithm.

Though many significant efforts have been done to handle each issue separately, few researchers have addressed the integration problem that combines search and attack issues. In terms of the search-attack integration issue, Hu *et al*. [21] developed a hierarchical solution framework to address the mission planning problem, however, the proposed method focused on optimal task assignment and ignored trajectory optimization. Kim *et al*. [22] presented a response threshold model (RTM) that had satisfactory flexibility, but obstacles

or threats were not considered for the environment. Zhen *et al*. [23], [24] proposed an intelligent self-organized mission planning algorithm to achieve collaborative search and decentralized attack of multiple UAVs, which considered various constraints and threats in the environment. However, this method failed to take into account the synergy between multiple UAVs, especially when performing attack tasks.

Motivated by these facts, we proposed a hybrid approach to accommodate the challenges imposed by cooperative integrated search-attack problems. In our study, the integrated problem is decomposed into two parts: centralized task assignment and distributed path planning. On the one hand, the center node of UAVs assigns the corresponding tasks for each UAV to achieve an optimal scheme that maximizes the detection target probability and attack capability. On the other hand, UAVs collect information from reconnaissance to construct a probability map, then the fuse information is obtained by interactions with their neighbors, which guides the UAVs movement.

Furthermore, the optimization algorithms, including traditional algorithms and heuristic algorithms, formulate the mission planning problem as finding an optimal solution while fulfilling a set of optimization indices and constraints. Many traditional algorithms, such as convex optimization [25], [26] and gradient descent [27], [28] are widely adopted in solving numerous optimization problems. For instance, Yang *et al*. [26] proposed a novel algorithm to rapidly generate the time-optimal trajectory by convex optimization. Chai *et al*. [27], [28] designed an improved gradient algorithm to solve the trajectory optimization problem. Although these deterministic algorithms can generate certain optimization results in polynomial time, its computational complexity is generally large, so it may not suit for solving multi-UAV search-attack mission planning problem.

Unlike traditional optimization algorithms, heuristic algorithms do not depend on the mathematical performance of the problem, have no strict requirements on initial values, and can effectively deal with complex optimization problems with high dimensions. Typical heuristic algorithms such as Particle Swarm Optimization (PSO) [29], Ant Colony Optimization [23], [24], Genetic Algorithm [6], and Brain Storm Optimization algorithm [30] have been widely adopted. Compared with other heuristic algorithms, PSO is easy to implement with fewer control parameters. However, it has the problem of low search efficiency and fast prematurity [31]. Recently, a novel optimization algorithm namely Pigeon-inspired Optimization (PIO) [31] was proposed and extensively applied, which was inspired by the special navigation behavior of the pigeon homing process. Two types of operators were introduced to enhance the searching abilities at different stages, depending on the magnetic field and landmarks, respectively. Compared with other swarm intelligence optimization methods, PIO is superior, in the field of task assignment, trajectory optimization, etc. However, the standard PIO still has some shortcomings such as fast prematurity and less population search span. Normally, these shortcomings are overcome by the improvement of population diversity and evolutionary strategies, because the plentiful population diversity contributes to expanding the candidate solution space while the suitable evolutionary strategies [32]–[36] are conducive to developing the algorithm. To satisfy the requirements of search-attack mission planning in a discrete environment, a dynamic discrete PIO method is proposed in this article, in which the dynamic concept is introduced to improve population diversity.

The main contributions of the proposed approach are as follows. First, a novel dynamic discrete PIO (D^2PIO) algorithm is developed for the cooperative search-attack joint mission planning problem of multiple UAVs with complex constraints. In comparison to other heuristic optimization algorithms, the D^2PIO method can provide the optimal solution for UAVs with the superior searching ability for global optima in a discrete environment. Next, an information fusion method is adopted to construct the probability map to guide the subsequent coordinated movements. The probability map is utilized to improve the efficiency of detection and attack. Then, in order to obtain continuous curvature paths application of UAVs, the flyable trajectories are generated by B-spline curves with maximum curvature constraints. Finally, the 3D dynamic flight processes are demonstrated by a visual simulation platform with a customized human-machine interface (HMI), which is built by utilizing the Unity 3D and Visual Studio 2017 C++ MFC tools.

The remainder of this paper is organized as follows. Problem description and formulation are described in Section II. In Section III, the D^2PIO algorithm of the search-attack mission planning problem is proposed. In Section IV, the interactive visual simulation platform is briefly introduced. The simulations are performed to verify the effectiveness of the proposed methods in Section V. Finally, Section VI concludes this article.

II. PROBLEM FORMULATION

A. Problem Description

The key to cooperative mission planning is how to execute tasks collaboratively through the UAVs perception of the environment and information interaction between each other and how to automatically adjust the behavior of each UAV to ensure stable flight in case of internal and external changes. During the whole mission period, the UAVs can make their own decisions based on the information from respective sensors and adjacent UAVs, so that they autonomously coordinate the movements and alternately conduct the search and attack tasks to achieve an optimal configuration.

The basic elements of the mission planning scenario can be formulated by a four-tuple $\{U, T, E, C\}$, where $U = \{U_1, U_2, \ldots, U_{Nu}\}$ denotes the set of Nu homogeneous UAVs. Let $T = \{T_1, T_2, \ldots, T_{Nt}\}$ be the set of Nt targets. E represents the battlefield environment variable set $E = \{E_1, E_2, \cdots, E_{Nr}\}$. The set C denotes constraints of multi-UAV cooperative search-attack mission planning mainly

involve flight constraints, task constraints, and environment constraints. The detailed descriptions of these constraints are given in this section.

Flight constraints C_F: Three kinds of constraints should be satisfied on the flight level including maneuverability constraint C_F^m, collision avoidance constraint C_F^c, and on-board resources constraint C_F^r. Let $\{\theta_i(t), \varphi_i(t)\}$ be a pair of the angle of UAV i at moment t, which are pitch angle and heading angle, respectively. Therefore, the first kind of constraints on the flight level is as follows:

$$\theta_i(t) \in [\theta_{\min}, \theta_{\max}] \tag{1}$$

$$|\varphi_i(t)| \leq \varphi_{\max} \tag{2}$$

$$V_i(t) \in [V_{\min}, V_{\max}] \tag{3}$$

where $\theta_{\min}$ and $\theta_{\max}$ denote the minimum and maximum restricted values of pitch angle in the vertical direction. $\varphi_{\max}$ denotes the maximum turning angle value. $V_{\min}$ and $V_{\max}$ denote the minimum and maximum velocity. Furthermore, collision is fatal for UAVs. In order to move safely, the collision avoidance constraint is as follows:

$$d_{ij}(t) \geq d_{\text{safe}} \left(i, j = 1, 2, \cdots, Nu; i \neq j\right) \tag{4}$$

where $d_{ij}(t)$ represents the distance between the UAV i and UAV j at time t, and d_{safe} represents the safe distance. The limited resource carrying capacity of UAVs (such as weapons) will influence the mission execution

$$w_i(t) \leq w_{\max}(i = 1, 2, \ldots, Nu) \tag{5}$$

where $w_{\max}$ is the maximum resource load of UAVs.

Task constraints C_T: For targets that require multiple attacks, it is reasonable for multiple UAVs to execute the strike operation synchronously. Therefore, the terminal impact angle, duration time, and appropriate allocation of resources are the crucial factors that affects the attack effects. Three constraints should be satisfied on the task level including heading angle on target constraint C_T^h, resources allocation constraint C_T^r , and arrival time constraint C_T^t. Let $\kappa_{ij}(t)$ be the heading angle on target j of UAV i at moment t, respectively, therefore, the first constraints C_T^h is as follows:

$$\begin{aligned} &\left|\kappa_{ij}(t)\right| \in [\kappa_{\min}, \kappa_{\max}] \\ &i \in \{1, 2, \cdots, Nu\}, j \in \{1, 2, \cdots, Nt\} \end{aligned} \tag{6}$$

where $\kappa_{\min}$ and $\kappa_{\max}$ represent the minimum and maximum restricted values, respectively. The UAVs that are assigned to jointly attack the same target should satisfy the resources requirement, which is illustrated as follows:

$$\sum_{i=1}^{N_j} w_i - w_{\text{req}}^j \geq 0 \tag{7}$$

where N_j and w_{req}^j represent the required amount of UAVs and resource for target j. Suppose the straight length between the current position of UAV i and the target j is L_{ij}, the range of time can is as follows:

$$t_i \in \left[t_i^{\min}, t_i^{\max}\right] = \left[\frac{L_{ij}}{V_{\max}}, \frac{L_{ij}}{V_{\min}}\right]. \tag{8}$$

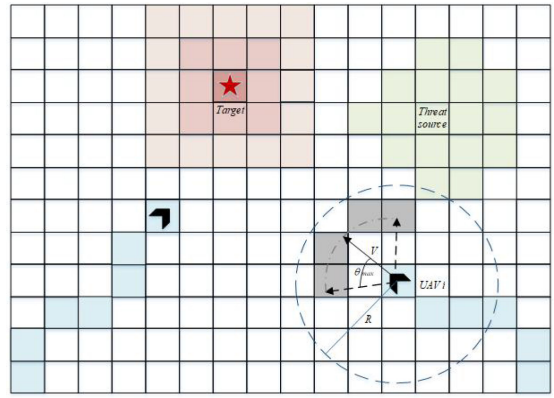

Fig. 1. Illustration of the search-attack mission area.

Therefore, the arrival time constraint is obtained as follow:

$$\bigcap_{i \in TR_j} \left[t_i^{\min}, t_i^{\max}\right] \neq \varnothing \tag{9}$$

where TR_j represent the set of N_j UAVs that are assigned to the target j. The notation $\varnothing$ stands for an empty set. In other words, these UAVs belonging to TR_j will be able to arrive near the target j within time constraints.

Environment constraints: From the perspective of safe flight, it is necessary to avoid the threats and obstacles in the hazardous environment. Thus, the constraint is given as follows:

$$\mathbf{x}_i(t) \cap \Xi = \varnothing \tag{10}$$

where $\mathbf{x}_i(t)$ represents the coordination of UAV i at moment t. Ξ represents the space occupied by threats and obstacles.

Since the threat is generally modeled as a hemispherical sphere, it can be expressed as

$$\begin{aligned} &\left\|\mathbf{x}_i(t) - \mathbf{x}_j^{\text{threat}}\right\| \geq R_j \\ &\left(i = 1, 2, \ldots, Nu; j = 1, 2, \ldots, Nt\right) \end{aligned} \tag{11}$$

where $\mathbf{x}_j^{\text{threat}}$ represents the location of threat j. $R_j$represents the impact range of the threatening hemisphere j.

B. Hybrid Mission Planning Architecture

Multi-UAV cooperative search-attack mission planning is a complex optimization problem, which aims at discovering and destroying the targets as many as possible with various constraints. As mentioned above, we study a group of fixed-wing UAVs performing search and attack targets over battlefield environments. Assume that:

1) the coordination of obstacles and threats is known;

2) each UAV can carry out both search and attack tasks;

3) target convoys are slowly moving in the mission area which can be regarded as a static target.

Multiple UAVs are deployed into a discrete rectangular region with length L and W shown as Fig. 1, where each UAV can perform both search and attack tasks for stationary or moving targets whose initial positions are unknown. The

targets that appeared within detection rang R can be discovered by the UAVs. The constraints on the turning angle limit the possible positions at the next moment. Considering the maximum turning angle and displacement per unit time, the gray grids in Fig. 1 denote the next allowable positions of the UAVs. When carrying out the reconnaissance tasks, the goal of the searching mission is to minimize the uncertainty of mission area in order to detect most targets as quickly as possible. When performing the attacking tasks, the UAVs work in an organized manner to destroy the target assigned by the mission decision layer.

III. METHOD FOR MISSION PLANNING

A. D^{2}PIO Applied to Optimize Mission Planning

PIO [31] is a novel swarm intelligence optimization method that simulates the special pigeon behavior. The magnetic field and the sun information is utilized at the earlier stage of their journey home, and landmarks information is utilized as getting closer to the destination. Their flight path is modified in time to ensure that they can take the optimal one to their destination during their journey. These two unique navigation operators of pigeons are formulated in the PIO algorithm, namely map and compass operator and landmark operator. The first is adjusted during the first part of optimization and another operator is used when the population is close to the global optima. Therefore, the position of each pigeon is regarded as a feasible solution, so the destination is considered to be the optimal solution of the problem.

The map and compass operator imitates the sun and the earth's magnetic field during pigeons flying. As getting closer to the destination, they will gradually reduce the dependence on the sun and the magnetic particles. The new position and velocity of pigeon i at the tth iteration can be calculated with the follows:

$$V_i(t) = V_i(t-1) \cdot e^{-Rt} + r_1 \cdot (X_g - X_i(t-1)) \tag{12}$$

$$X_i(t) = X_i(t-1) + V_i(t) \tag{13}$$

where $V_i(t)$ and $X_i(t)$ denote position and velocity of pigeon i at the tth iteration, respectively. R is the map and compass factor, which can be adjusted according to the problem. r_1 is a random number with a range [0, 1], and X_g is the current global best position, which is obtained by comparing the fitness value of all positions.

The landmark operator imitates the function of regular landmarks. When being close to the destination, they will rely on adjacent landmarks. Pigeon in the best position in the current iteration is considered as the intermediate destination. In the landmark operator, half of pigeons are far from the intermediate destination, so the population size will be reduced per iteration as follow:

$$N_P(t) = \text{ceil}\left(\frac{N_P(t-1)}{2}\right) \tag{14}$$

where $N_P(t)$ denotes the population size at the tth iteration. In addition, ceil$(\cdot)$ is a rounding function.

The center position of the population at the tth iteration is expressed as

$$X_c(t) = \frac{\sum X_i(t) \cdot \text{fitness}\,(X_i(t))}{N_P \sum \text{fitness}\,(X_i(t))} \tag{15}$$

where fitness$(\cdot)$ is the objective function to evaluate the quality of each pigeon individual. The new position of pigeon i at the tth iteration can be calculated as follows:

$$X_i(t) = X_i(t-1) + r_2(X_c(t) - X_i(t-1)) \tag{16}$$

where r_2 is a random number between 0 and 1.

Similar to PSO, the PIO algorithm is also a stochastic optimization algorithm constructed by simulating the behavior of natural biological groups. The velocity and position of PSO are updated as following [30]:

$$V_i(t) = \omega \cdot V_i(t-1) + c_1 \cdot r_1 \cdot (X_g - X_i(t-1)) + c_2 \cdot r_2 \cdot (X_p - X_i(t-1)) \tag{17}$$

$$X_i(t) = X_i(t-1) + V_i(t) \tag{18}$$

where ω is the inertia coefficient. c_1 and c_2 are the acceleration coefficients. X_p is the local best position. From the equations, it can be seen that this algorithm tracks the best global fitness of all the particles and the best fitness achieved by the particle. However, two independent stages are designed in PIO according to the homing rules of the pigeons. Compared with PSO, PIO has the advantages of stronger global search ability, faster calculation speed, and avoiding premature.

In order to improve the shortcomings of the standard PIO, the D^{2}PIO algorithm is proposed. The D^{2}PIO is based on the basic PIO algorithm formula shown in (12)–(16) but with some minor modifications. Inspiring from the researches presented in [29], the local optima detector LODg for global best position and LODc for the center position are adopted in two stages of PIO algorithm, respectively. These two detectors count the number of consecutive iterations without amelioration in the global best and center position so that external thrust is provided to boost pigeons toward unexplored space if they are saturated. Thus, the stagnation and falling local optimum problems can be avoided in this way.

Dynamic concept [29]: Our work includes implementing a dynamic version of PIO. It acquires its dynamic aspect from the parameters (Sg, Sc) given by the user. When the global optima X_g is not improved for a user-defined threshold, i.e., LOD$g = Sg$, X_g will be restructured to $\hat{X}_g$ given in (19). Respectively for center position X_c in landmark operator, i.e., LOD$c = Sc$, that will be restructured to $\hat{X}_c$ given in (20)

$$\hat{X}_g = \min\left(X_{n_1} + \frac{n_1}{n_1 + m_1} \cdot (Xg\_\text{hist}_{m_1} - X_g), X_g\right) \tag{19}$$

$$\hat{X}_c = \min\left(X_{n_2} + \frac{n_2}{n_2 + m_2} \cdot (Xc\_\text{hist}_{m_2} - X_c), X_c\right) \tag{20}$$

where n_1 and n_2 both are random numbers from 1 to population size. Xg_hist and Xc_hist denote the historical values of global optima X_g and center position X_c, respectively.

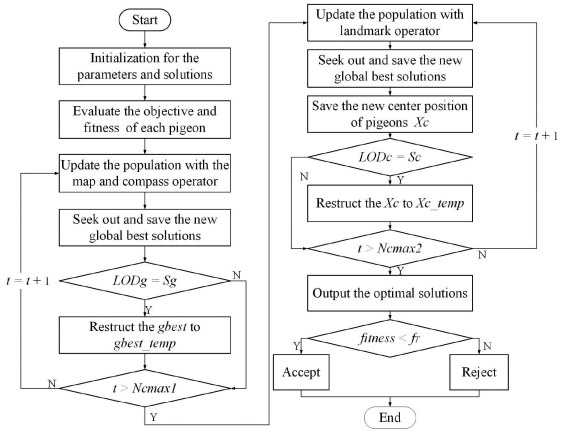

Fig. 2. Flow chart of task assignment using D^{2}PIO.

m_1 and m_2 are random numbers that meet the size range. $c = \min(a, b)$ means that c is equal to the minimum of a and b.

Discrete concept [37]: Because the values in population are integers, we usually round these numbers to the closest task number by dropping the fractional part. Thus, we yield it as $\{1, 0, 6, 1, 2\}$ if the pigeon position is calculated as $\{1.2, -2.1, 6.4, 0.2, 1.6\}$. Make this adjustment for all pigeons and the cycle of updating the pigeon position and searching for the global best until satisfactory results or termination criteria are met.

Solution acceptance strategy: Due to the high dynamic environment, the search-attack mission planning is real-time according to the changes in the environment, which may cause frequent task switching and unable to accomplish tasks effectively. Therefore, we propose a solution acceptance strategy, in which a dynamic threshold f_T is introduced. The new optimal solution can be accepted only when the cost of executing this plan below the current dynamic threshold f_T; otherwise, UAVs continue to perform the previous tasks.

Define the population size of the pigeons, the dimension of the parameter vectors and the computational cost of the objective function J as N_P, D, and L_f, respectively. The complexity of the two operators in the standard PIO algorithm can be described as $O(N_P(D + L_f))$ and $O(N_P \log N_P + D \log N_P + L_f \log N_P)$. In the proposed D^{2}PIO method, the computational cost of the discrete and dynamic mechanism are L_{di} and L_{dy}, respectively. The computational cost increases as $O(N_P(D + L_f + L_{di} + L_{dy}))$. Define the total iteration as N_c, the total computation cost of D^{2}PIO algorithm is $O(N_c(N_P \log N_P + DN_P + L_f N_P + L_{di} N_P + L_{dy} N_P))$.

Fig. 2 shows a flow chart of the D^{2}PIO method. Give the iteration thresholds for two respective operators Ncmax1 and Ncmax2. t denotes the current iteration count. Moreover, $g$best_temp and Xc_temp are calculated according to (17) and (18), respectively.

B. Objective Function

Suppose that each UAV makes its own decisions with a separate processor for the search-attack mission, then the individual objective function is designed as follows:

$$J_i = \omega_i \cdot J_i^s + (1 - \omega_i) \cdot J_i^a \tag{21}$$

where ω_i indicates Boolean variable, $\omega_i = 1$ means UAV i is performing the search task, otherwise, UAV i is performing attack task.

The global objective function J is formulated as:

$$J = \sum_{i=1}^{Nu} \left(\omega_i \cdot J_i^s + (1 - \omega_i) \cdot J_i^a\right) \quad \text{subject } C. \tag{22}$$

1) *Construction and Updates of Probability Map:* Traditional target search methods such as the spiral and lawn mower methods both are simple but inefficient. Particularly, these methods may underperform and even fail in a dynamic environment [5]. In order to achieve better performance and efficiency, the consensus-based probability graph method is adopted. The battlefield environment can be described as a grid-based probability cell which corresponds to a discrete search area with an associated probability target.

To construct the probability map of an uncertain even unknown area, multiple UAVs collect information through collaborative detection and fuse information through interaction with their neighbors. The UAVs' following cooperative motion is guided by the goal that maximizes the probability that the target is detected. In this section, the target distribution is estimated consistently by using the Bayes rules. The probability map is established by combining both current detection information of their neighbors and the historical estimations based on consensus algorithm, which is associated with the uncertainty.

Individual Probability Map Updating: The individual probability map of each UAV is updated by using the Bayes rules. UAV i constructs its individual map according to the observation $Z_i^c(t)$ in cell c at time t

$$P(\tau_c = 1 | Z_i^c(t)) = \frac{P(Z_i^c(t) \big| \tau_c = 1) P(\tau_c = 1)}{P(Z_i^c(t))} \tag{23}$$

where $\tau_c = 1$ indicates the event that a target appears in cell c and otherwise $\tau_c = 0$. $P(\tau_c = 1)$ denotes the required prior probability of the target state in cell c. It is feasible to use the history estimations of the probability at the previous time as the prior probability. Combining sensor information $Z_i^c(t)$ and prior probability $P_i^c(t-1)$, probability, $P_i^c(t)$ is estimated as follows:

$$P_i^c(t) = \frac{P(Z_i^c(t) \big| \tau_g = 1) P_i^c(t-1)}{P(Z_i^c(t))}. \tag{24}$$

Using the law of total probability, the probability estimate can be expressed as follow:

$$\begin{aligned} P(Z_i^c(t)) = {} & P(Z_i^c(t) \big| \tau_c = 1) P(\tau_c = 1) \\ & + P(Z_i^c(t) \tau_c = 0) P(\tau_c = 0) \end{aligned} \tag{25}$$

where $P(Z_i^c(t) | \tau_c = 1)$ and $P(Z_i^c(t) | \tau_c = 0)$ denote the probability of occurrence for an observation result under certain target-present situations. Detection probability $p_c = P(Z_i^c(t) = 1 | \tau_c = 1)$ and false alarm probability $p_f =$

$P(Z_i^c(t)=1|\tau_c=0)$ are two characterized performance parameters of sensors, where $0<p_c<1$ and $0<p_f<1$. Therefore, the individual probability map is written as

$$\begin{aligned} P_i^c(t) &= P(\tau_c=1|Z_i^c(t)) \\ &= \frac{P(Z_i^c(t)|\tau_c=1)P_i^c(t-1)}{P(Z_i^c(t)|\tau_c=1)P(\tau_c=1)+P(Z_i^c(t)|\tau_c=0)P(\tau_c=0)} \\ &= \begin{cases} \frac{p_c P_i^c(t-1)}{p_c P_i^c(t-1)+p_f(1-P_i^c(t-1))} & Z_i^c(t)=1 \\ \frac{(1-p_c)P_i^c(t-1)}{(1-p_c)P_i^c(t-1)+(1-p_f)(1-P_i^c(t-1))} & Z_i^c(t)=0 \\ P_i^c(t-1) & \text{otherwise} \end{cases} \end{aligned} \tag{26}$$

Cooperative probability map updating: The updating process of the individual probability map has been formulated in the previous section by using the Bayes rule. UAVs can update their probability maps through information sharing to promote search efficiency. The UAVs fuse the probability map by incorporating its own information and the received one from its neighbors, expressed by

$$Q_i^c(t)=\sum_{j=1}^{Nu}\lambda_j P_j^c(t) \tag{27}$$

where λ_j is the weight. Therefore, the uncertainty of the environment can be expressed as the Shannon comentropy of the targets existence probability, given by

$$\begin{aligned} \chi_i^c(t) &= H(Q_i^c(t)) \\ &= -Q_i^c(t)\log_2 Q_i^c(t)-(1-Q_i^c(t))\log_2(1-Q_i^c(t)). \end{aligned} \tag{28}$$

Guided by the uncertainty χ, UAVs tend to fly toward locations with high uncertainty, which, in turn, reduces the uncertainty by updating the sensor observations. Then UAVs repeat this process until uncertainty distributions in the mission area fall below a predetermined threshold.

The search task goal is to reduce the uncertainty of the mission area so as to maximize the target occurrence detection probability. Let $\mathbf{x}_i^{\max}$ be the location with the highest uncertainty of UAV i. Therefore, the search cost is expressed by

$$J_i^s(t)=\frac{1}{L}\left\|\mathbf{x}_i^{\max}(t)-\mathbf{x}_i^c(t)\right\| \tag{29}$$

where $\mathbf{x}_i^c$ indicates the candidate cells of UAV i at moment t. L denotes the length of the mission area.

2) *Target Allocation Based on Response Threshold Sigmoid Model:* When UAVs detect one or more targets, the target assignment module provided the optimal solution that suitable UAVs are allocated to perform the attack task. An attack task along with target information is announced with its neighbors, including the type, location, munitions requirement, etc. To check the eligibility of each UAV that receives the attack task information, the RTM [39] is employed to calculate the attack probability of each UAV. In RTM, the attack task of the UAV is influenced by the external stimulus and the response threshold. Whether or not the UAV holds a proper attack capability with current munition resources under the external stimulus, and if so, respond to the attack task with a probability. The external stimulus of an attack task $S_i(t)$ is simply determined as a corresponding target's value at the time. However, the UAVs may not respond to realize effective target assignments in the dynamic environment timely based on the basic RTM. Therefore, the Sigmoid function is utilized with the exponential term to make the UAVs respond quickly to the changes. The response threshold $\theta_{ij}(t)$ of UAV i to attack target j is given as follow:

$$\theta_{ij}(t)=\alpha\cdot\Delta t_j+\beta\cdot L_{ij} \tag{30}$$

where Δt_j denotes the maximum arrival time lag between multi-UAV team that is assigned to attack target j. L_{ij} represents the distance between UAV i and target j. α and β the corresponding weight coefficient.

The state transition probability based on response threshold sigmoid model (RTSM) is rewritten as follow:

$$\tilde{P}_{ij}^f=\begin{cases} \frac{1}{1+e^{n(\theta_{ij}-S_i(t))}} & S(t)>S_0 \\ 0 & \text{otherwise} \end{cases} \tag{31}$$

where $\tilde{P}_{ij}^f$ is the state transition probability which determines whether the UAV i begin to attack target j. n determines the slope of the probability function. The parameter n is generated randomly for each UAV [39]. Additionally, one UAV with a larger n is more sensitive to the environmental changes and can respond rapidly to the stimulus and threshold.

We interpret this probability as the preference level of UAV i to select target j. Since one UAV is given multiple candidate grids, the UAV needs to pick only one among them for the next time position under various constraints. Thus, a probability for UAV i to select target j is given by a normalized probability using the preference level as

$$P_{ij}^f(t)=\frac{\tilde{P}_{ij}^f(t)}{\sum_{k\in CG(i)}\tilde{P}_{ik}^f(t)}. \tag{32}$$

According to the function, one target is more likely to be chosen if its preference level is higher. The goal of the attack task is to maximize the attack preference probability. Therefore, it is expressed in the minimization problem as follow:

$$J_i^a(t)=\frac{1}{P_{ij}^f(t)+\varepsilon}. \tag{33}$$

C. B-Spline Curve for Path Smoothing

In the aforementioned path planning method, curvature discontinuities of the discrete waypoints may lead to poor quality paths even instability of the over harmonic control system. In order to obtain continuous curvature paths application of UAVs, Dubin's curves are widely utilized to path smoothing in planar and 3D scenarios [40]. However, computing the appropriate Dubin's set is pointed to be challenging in uncertain even unknown environment [41]. Therefore, the B-spline curves are used to smooth the UAVs' path due to the robustness and practicality, which can allow UAVs to resolve their constraints efficiently.

A pth degree B-spline curve $c(u)$ is defined by n control points and a knot vector $\hat{u}$. The number of knots m is equal to $n+p+1$. The knot vector $\hat{u}$ consists of m non-decreasing

real numbers, and u is a normalized curve length parameter, given as follows [41]:

$$c(u) = \sum_{i=0}^{n} N_{i,p}(u) P_i \tag{34}$$

where P_i denotes the ith control point. $N_{i,p}(u)$ represents the ith B-spline basis function of a p-degree curve. Each basis function performs local control at that specific point. Equations (35) and (36) define recursive algorithm for calculating basis functions. The corresponding knot vectors, $\hat{u}_i$ and $\hat{u}_{i+1}$, are used to evaluate the basis functions. Using the recursive method, higher order basis functions can be calculated

$$N_{i,0} = \begin{cases} 1 & u \in [\hat{u}_i, \hat{u}_{i+1}) \\ 0 & \text{else} \end{cases} \tag{35}$$

$$N_{i,p}(u) = \frac{u - \hat{u}_i}{\hat{u}_{i+1} - \hat{u}_i} N_{i,p-1}(u) + \frac{\hat{u}_{i+p+1} - u}{\hat{u}_{i+p+1} - \hat{u}_{i+1}} N_{i+1,p-1}(u). \tag{36}$$

Define the fitting error as follows:

$$SSE = \sum_{j=1}^{m} \left\| c_j - C(u_j) \right\|^2 = \sum_{j=1}^{m} \left\| c_j - \sum_{i=0}^{n} N_{i,p}(u_i) P_i \right\|^2. \tag{37}$$

The B-spline curves under different number of control points are shown in Fig. 3 when the UAV's turning angle is 60° (max turning angle). The results show that with the increase of the number of control points, the B-spline curve becomes closer and closer to the broken line segment (original path) shown in Fig. 3(a). As shown in Fig. 3(c), the increase in the number of control points can reduce the fitting error. In addition, the curvature corresponding to the normalized path length increases as the number of control points shown in Fig. 3(b). However, choosing the appropriate number and position of control points can not only ensure that the smooth trajectory can meet the requirements of precision but also meet the constraints of turning radius.

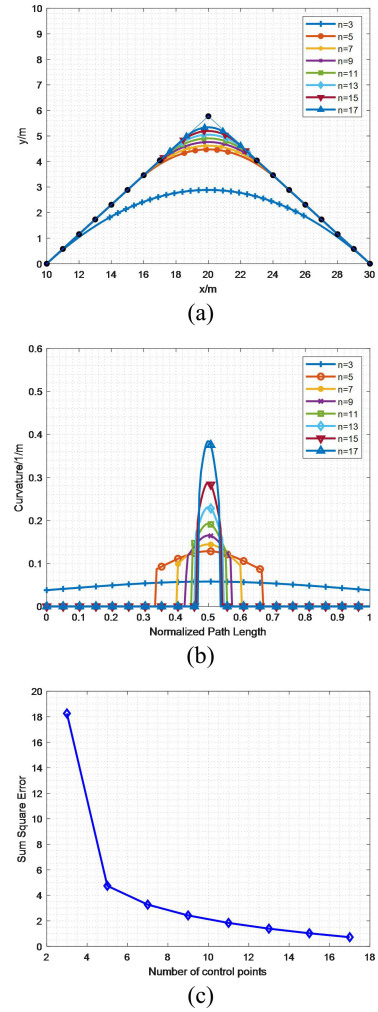

Fig. 3. Effect of control points on B-spline curves. (a) B-spline curves with a different number of control points. (b) Curvature of B-spline curve with different number of control points. (c) Fitting error of B-spline curve under different control points.

IV. VISUAL SIMULATION PLATFORM

The visual simulation platform is in allusion to the need for multi-UAV cooperative search-attack mission planning, using customized human-machine interaction interface based on MFC tools in Visual Studio 2017, user datagram protocol (UDP) communication method, and basic 3D visual software. First, the corresponding instructions or data are received from the HMI and transmitted to the background computation layer and visual software synchronously. Second, the instructions/data are processed by the data resolving module. Then, the simulation data of flight position and attitude are calculated via executing the arithmetic module and transmitted through the UDP network. Finally, the driving of the visualization engine in basic visual software is to realize a 3D visual display of multi-UAV flight simulation. Based on the above ideas, build the overall architecture of the visual simulation platform.

The HMI is developed by Microsoft Foundation Classes (MFC) in Visual Studio 2017. MFC is a C++ class library implemented by Microsoft which encapsulates most of the Windows API functions and uses object-oriented methods to invoke Windows API and develop applications more easily. Many common functions are automated and customized application frameworks are provided, such as document framework view structures and active documents. Meanwhile, the auxiliaries for MFC such as class wizard reduce software development time.

Create an MFC Application project called "VRScene" in Visual Studio 2017, then add the different controls in the project and write the corresponding response codes, so as to generate the executable HMI as shown in Fig. 4. Important controls applied in HMI include Button Control, Edit Control, Combo Box, and TeeChart Charting Controls. Typical buttons are the checkbox, radio button, and pushbutton. This Button Control is used to respond to user mouse clicks and process accordingly through message maps ON_BN_CLICKED such as:

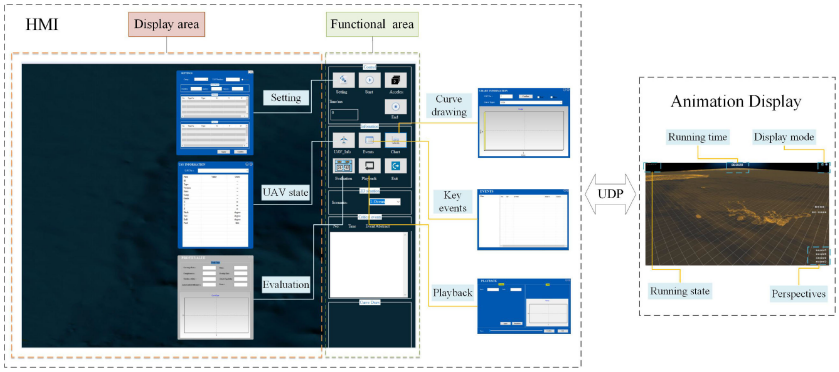

Fig. 4. Visual simulation platform.

```
void CVRSceneView::OnBnSetting()
{
SettingDlg *pDlg = new SettingDlg;
pDlg->Create(IDD_SETTING_DLG,this);
pDlg->ShowWindow(SW_SHOW);
}
```

The Edit Control and Combo box provide a programming interface for formatting text. Unlike Edit Control, a Combo Box has more abundant functions, which consists of a list box combined with either a static control or edit control. The text can be obtained or set by the functions *GetDlgItem*() and *SetItem*(), respectively. In addition, the curves are drawn through TeeChart Charting Controls, which provides an excellent universal component suite to meet a variety of charting requirements.

The HMI can provide users with a convenient way to operate the simulation system. Users can see not only the dynamic three-dimensional effect of multi-UAV mission planning but also specific data in various forms. The main interface consists of two parts: functional area and two-dimensional display area as shown in Fig. 4. The functional area is able to control the running and stopping of data computation, initialize the parameters used in mission planning, display the specific data in various forms, and evaluate the performance of mission execution, etc. The flight trajectories of UAVs are drawn synchronously in the two-dimensional display area.

Two computers are networked to accomplish simultaneous visualization simulation of multi-UAV flight and task execution process in different forms, such as three-dimensional animation and two-dimensional curves dynamic drawing. UDP is adopted to transfer data between the two computers. UDP is a connectionless transport layer protocol that provides simple and unreliable transaction-oriented information delivery services. In addition, sending a string requires a specific information format. The basic data transmission format mainly includes the coordinates information, attitude information, and event information.

TABLE I
Configurations for D^{2}PIO With Different Parameters

Case 1		Case 2		Case 3		Case 4	
N_p	5~100	N_p	100	N_p	100	N_p	100
R	0.5	R	0.25~0.7	R	0.5	R	0.5
Sg	5	Sg	5	Sg	5~15	Sg	9
Sc	7	Sc	7	Sc	7	Sc	7~19

V. SIMULATION RESULTS AND ANALYSIS

A. Comparison Analysis of Optimization Algorithm

In order to select appropriate parameter values in the subsequent application of the D^{2}PIO algorithm, we analyze the effect of the parameter variations on the D^{2}PIO algorithm shown as Fig. 5, namely population size N_p, the map and compass factor R, dynamic threshold Sg, and dynamic threshold Sc. The initial configurations of the four cases are shown in Table I. In addition, the fitness mean values are obtained after 100 executions to ensure reliability.

As shown in Fig. 5, all these four parameters affect the performance of the algorithm to varying degrees. In Fig. 5(a), the increase in population enriches the diversity of solution space, so as to improve the performance. However, too large population size may result in an increase in the cost of computing resources. Moreover, in Fig. 5(b))–(d), the algorithm performance is better only when the parameter value is moderate. Improper values of these parameters may have detrimental consequences.

To prove the effectiveness and superiority of the D^{2}PIO algorithm, the objective of this subsection is to compare the optimization results with other existing methods, including basic PSO (BPSO), basic PIO (BPIO), dynamic PSO (DPSO) [29], and MPSO [41]. The initial parameters settings of the candidate optimization algorithms are shown in Table II. The parameters in MPSO are the same as those in [41] except for MFD$*$. The dimension $D = 10$, which means 10 UAVs are to be optimized. Additionally, to make a fair comparison between candidate algorithms, the fitness mean value is utilized to estimate their performance after 100 executions.

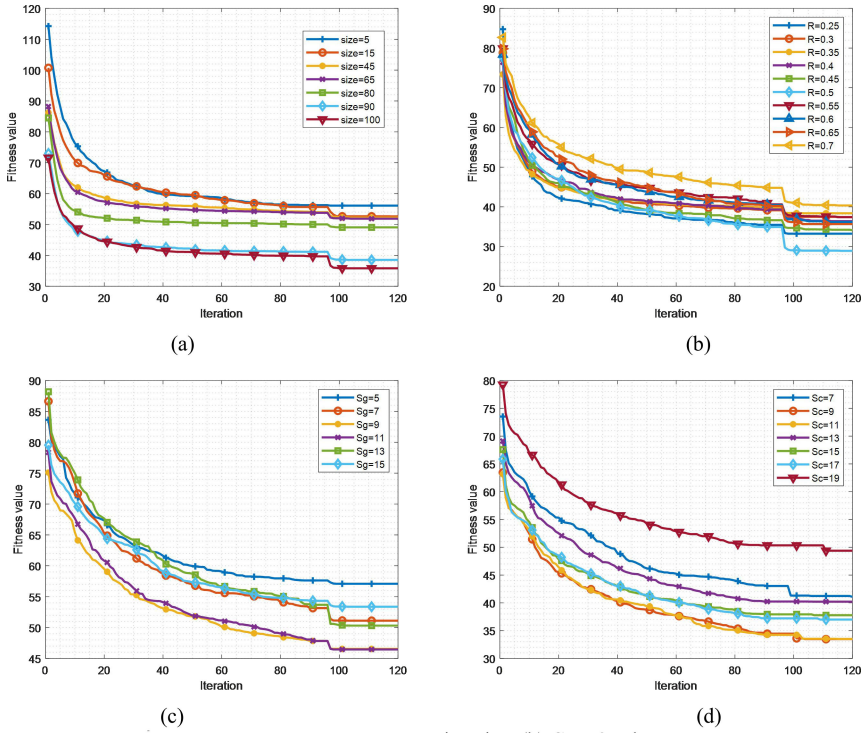

Fig. 5. Effect of parameters on D^{2}PIO algorithm (a) Case 1: Population size. (b) Case 2: The map and compass factor. (c) Case 3. Dynamic threshold *Sg*. (d) Case 4: Dynamic threshold *Sc*.

TABLE II
Configurations for Candidate Algorithms

BPIO and D^{2}PIO Parameters			
Ncmax1	80	N_p	100
Ncmax2	40	*Sg*	9
R	0.5	*Sc*	11
BPSO, DPSO and MPSO Parameters			
Ncmax	120	ω	0.9
c_1	2	c_2	2
N_p	100	*Sg*	9
*MFD**	7		

TABLE III
Target and Threat Information

Target Label	Location/m	Value	Threat Label	Coordinate/m
1	(2285,6,0),	0.4647	1	(-5000,10,0)
2	(1670,-2991,0)	0.5245	2	(-3000,3500,0)
3	(-1919,-2026,0)	0.5141	3	(-1000,1800,0)
4	(4110,2945,0)	0.5667	4	(2000,3000,0)
5	(-986.87,303.06,0)	0.5748	5	(3000,-2500,0)
6	346.59,2279.46,0	0.4821	6	(800,-1800,0)

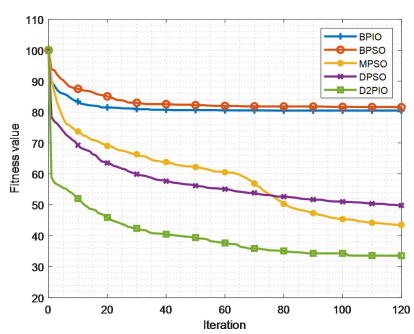

Fig. 6. Comparison evolution curves.

Fig. 6 shows the relationship between the fitness function and iteration count of D^{2}PIO and four other optimizers. The fitness value represents the total consumption of search-attack mission planning for 10 UAVs. The smaller the value is, the better the performance of multiple UAVs carrying out the tasks. Obviously, the proposed D^{2}PIO algorithm is the best optimization algorithm, while the four other algorithms do not achieve good results, since they possibly locate into the local optimum, especially BPIO and BPSO.

B. Mission Execution Analysis

The mission area (10×10) km^2 is divided into the discrete grids 250×250. There scatter six targets and six threats in the area, whose initial positions are given in Table III. However, the information on the targets is uncertain even unknown for the UAVs.

Multiple UAVs are deployed to the mission area. The initial position, heading angle, and relative task parameters are given in Table III. All UAVs are initially located near the position $(-4500, 3000, 400)$ m. The UAV's maximum turning angle $\varphi_{\max} = 60^\circ$, the moving speed $V \in [45\,\text{m/s}, 60\,\text{m/s}]$ (1–2 grids per unit time), the detection radius $Rs = 800$ m (20 grids) and the attack radius $Ra = 400$ m (10 grids). The detection probability $Pc \in [0.8, 1)$ and false alarm probability $Pc \in [0.05, 0.1]$. Each UAV is strictly limited

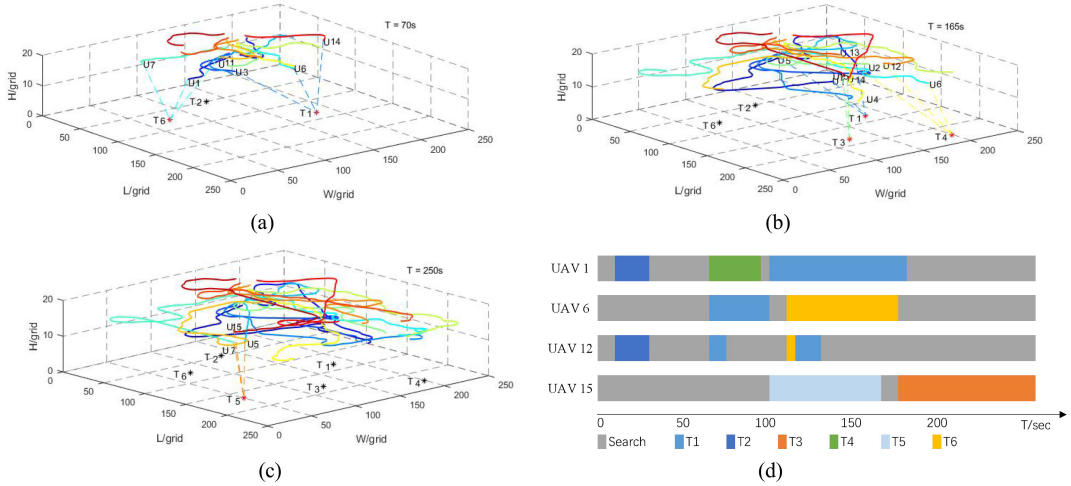

Fig. 7. Cooperative search-attack mission planning results in Case 1. (a) Submission in the second 70. (b) Submission in the second 165. (c) Submission in the second 250. (d) Task loads of UAV1, UAV6, UAV12, and UAV15.

to perform three attack tasks due to its onboard resources restraint. Additionally, the simulation step size is fixed as 5 seconds per step in this section.

It is challenging to organize the UAVs to conduct the cooperative search-attack mission under various constraints, especially in a complex environment with threats. In order to analyze the feasibility and effectiveness of the proposed method, the following four cases are set. Case 1 and Case 2 are designed as non-threatening and threatening scenes, respectively. In addition, the total resources on the UAVs are sufficient for the mission, which can verify the feasibility and reliability of the proposed method under ideal conditions. Different from the previous cases, Case 3 is designed as an insufficient resources scene to comprehensively analyze the applicability of the algorithm. Finally, Case 4 serves as a comparison of Case 1, illustrating the rationality and effectiveness of the solution acceptance strategy.

CASE 1: Fifteen UAVs are deployed into the mission area, where six targets are scattered in the discrete environment without obstacles or threats. Fig. 7(a)–(c) shows that the multiple UAVs perform on the mission at different moments. As shown in Fig. 7(a), the black asterisk, i.e., target 2 has been successfully destroyed. Meanwhile, target 1 and target 6 (the red asterisks) had been discovered and two teams are formed to attack these two targets in the second 70. Fig. 7(b) shows that three teams are assigned to attack these three discovered targets in the second 165. As shown in Fig. 7(c), the team {UAV5, UAV7, and UAV15} is close to target 5 to attack it. Then, the multiple UAVs finish the search-attack mission in the second 255. Fig. 7(d) illustrates the task load of partial UAVs during the whole search-attack mission planning period. In this case, it is difficult to plan the search path and perform the cooperative attack tasks in an uncertain environment. However, the mission can be quickly and effectively completed in a threat-free scene, which verifies the basic feasibility of the proposed method.

TABLE IV
Mission Performance

	Case 1	Case 2	Case 3	Case 4
Completeness	100%	100%	50%	0%
Time/Sec	255	300	--	--
Flight distance/km	13.81	13.99	23.02	13.02
Average task loads	1.53	1.67	3.60	4.67

CASE 2: Fifteen UAVs are deployed into the mission area, where six unknown targets and six known threats are scattered in the discrete environment. The existence of threats greatly restricts the motion of UAVs, increasing the complexity of mission planning. Fig. 8(a)–(c) shows that the multiple UAVs perform on the mission at different moments. In the second 70, target 2 has been successfully destroyed shown in Fig. 8(a), and two teams are close to target 1 and target 6, respectively. Fig. 8(b) shows that two teams are assigned to attack the discovered target 4 and target 5 in the second 165. One consists of UAV1, UAV4, and UAV5. Another consists of UAV2, UAV3, and UAV13. After flying 330 seconds, the UAV3, UAV7, and UAV13 cooperatively destroy the last target shown in Fig. 8(c). Fig. 8(d) illustrates the task load of partial UAVs during the whole search-attack mission planning period. In this case, the UAVs complete the search-attack tasks while avoiding obstacles, illustrating the practicability of the proposed method in a complex environment.

CASE 3: Five UAVs are deployed into the mission area, where six targets and six known threats are scattered in the discrete environment. Too few UAVs, that is, lack of resources for mission execution will lead to poor performance. Meanwhile, due to the existence of threats, more resources and time are needed to ensure safety. Fig. 9(a)–(c) shows that the multiple UAVs perform on the mission at different moments. Since multiple UAVs are required to attack targets simultaneously, the search-attack

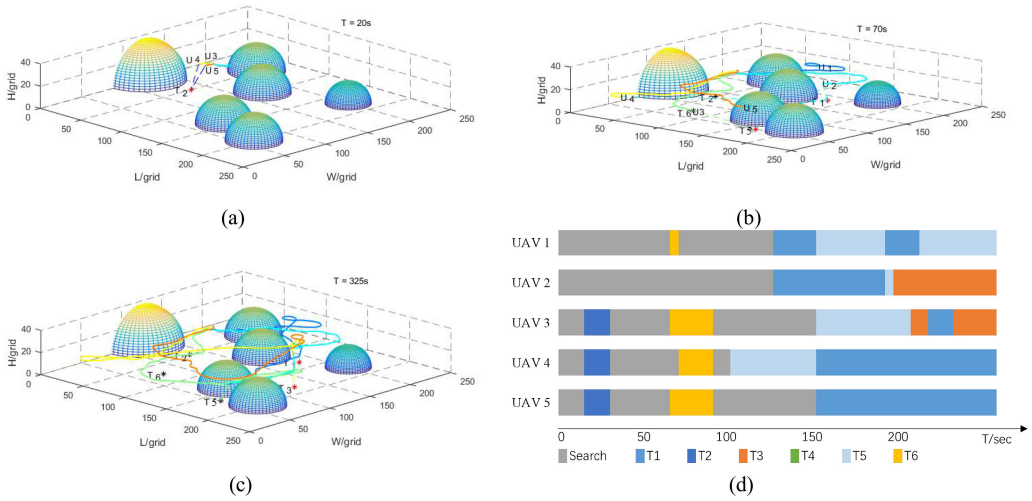

Fig. 8. Cooperative search-attack mission planning results in Case 2. (a) Submission in the second 70. (b) Sub-mission in the second 165. (c) Submission in the second 300. (d) Task loads of UAV1, UAV6, UAV12, and UAV15.

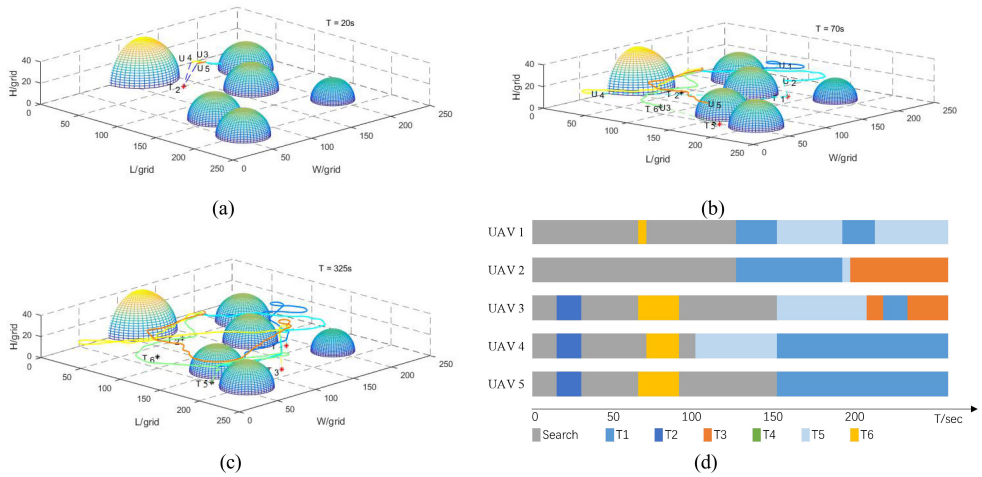

Fig. 9. Cooperative search-attack mission planning results in Case 3. (a) Submission in the second 20. (b) Submission in the second 70. (c) Submission in the second 325. (d) Task loads of UAV1, UAV6, UAV12, and UAV15.

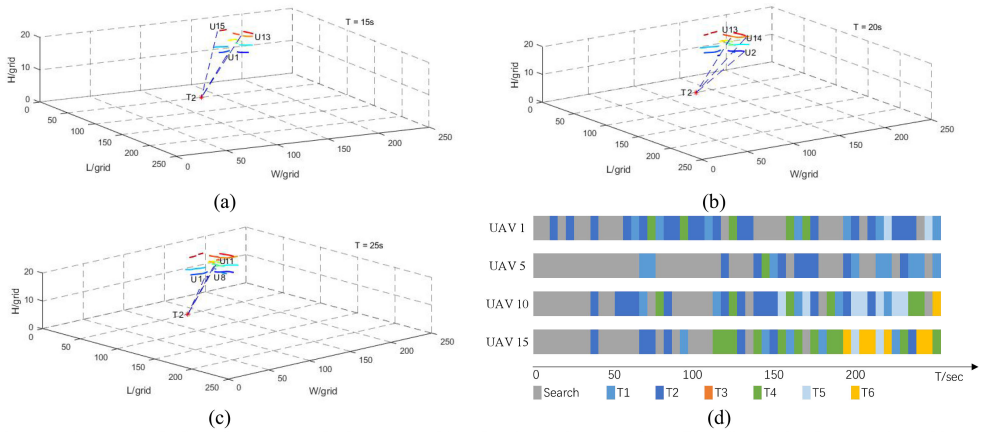

Fig. 10. Cooperative search-attack mission planning results in Case 4. (a) Submission in the second 15. (b) Submission in the second 20. (c) Submission in the second 25. (d) Task loads of UAV1, UAV6, UAV12, and UAV15.

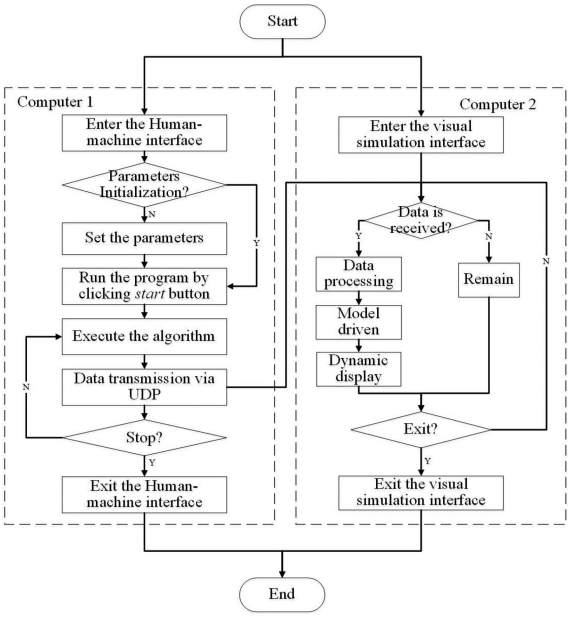

Fig. 11. Internal operation diagram of the simulation platform.

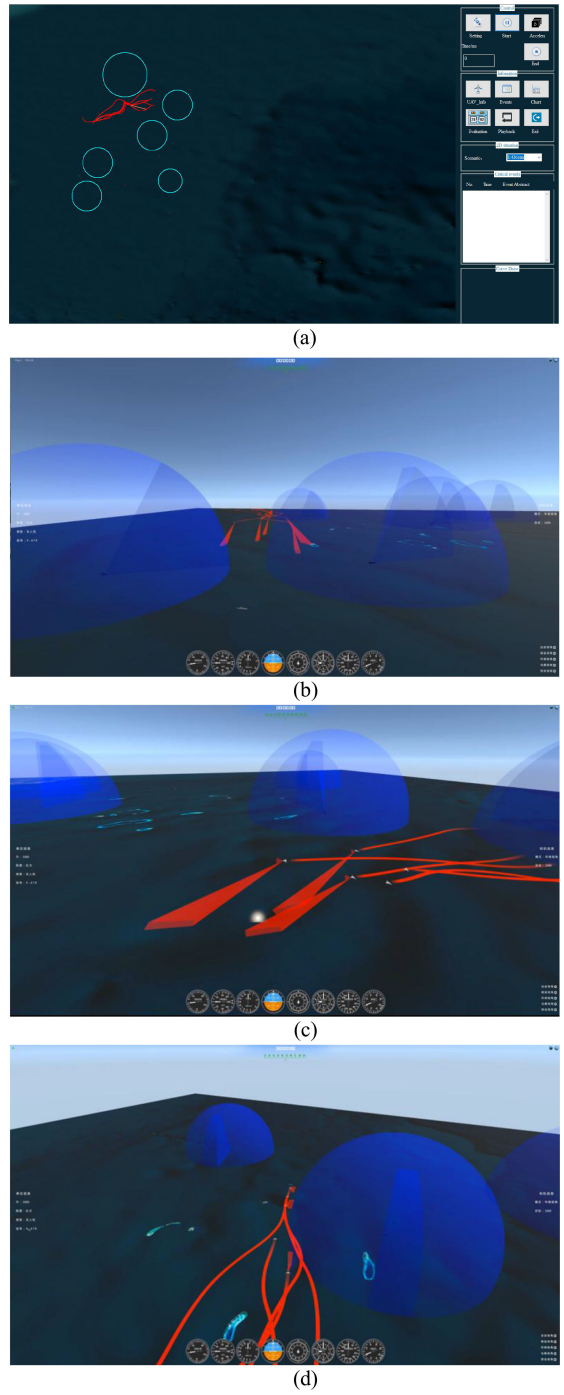

Fig. 12. Visual simulation platform snapshots. (a) Snapshot of HMI. (b) Snapshot of searching. (c) Snapshot of attacking. (d) Snapshot of threat avoidance.

mission does not be completed within the limited time due to the onboard resource restraints. As shown in Fig. 9(c), target 1 and target 3 have been spotted and assigned to the UAVs, but have not yet been destroyed successfully because of lack of resources. In addition, target 4 has not been discovered. Fig. 9(d) illustrates the task load of partial UAVs during the whole search-attack mission planning period. It is obvious that the mission payload is heavy for each UAV. It is obvious that the mission payload is heavy on each UAV due to insufficient resources. However, half of the mission completion is achieved using a small number of UAVs, showing the adaptability of the proposed method in extreme situations.

CASE 4: Fifteen UAVs are deployed into the mission area, where six unknown targets are scattered in the discrete environment without obstacles or threats. Since the scene changes dynamically, the mission planning scheme should also be adjusted accordingly. However, switching tasks too frequently may result in performance degradation and even failure to complete the tasks. In this case, the solution acceptance strategy is not adopted in the optimization algorithm. Fig. 10(a)–(c) shows that the multiple UAVs perform on the mission in a short period (within 10 s). Obviously, task switching is too frequent without the solution acceptance strategy compared to the previous three cases, which leads to the very poor performance of the mission. The task load of each UAV also illustrates this disadvantage intuitively shown in Fig. 10(d). Therefore, it is reasonable and necessary to adopt the solution acceptance strategy, which ensures that the task adjustment will not be too sensitive to environmental changes and effectively improves task completion.

In addition, four indicators in the above cases are compared, namely completeness, duration time, average flight distance, and average task loads between multi-UAV search-attack mission cases. The results are shown in Table IV, we can see that for the first three cases, the amounts of UAVs and the requirement of target attacking influence the completeness and duration time. This is mainly because of the resource restraint of UAVs. Compared with case 4, it is obvious that frequent task switching leads to the UAVs mission execution confusion and unable to complete the tasks within a limited time.

C Snapshots Using Visual Simulation Platform

Run the visual simulation platform in Windows 10 operating system. Two computers are used to display the human–computer interaction interface and the dynamic presentation interface, respectively. Through the interactive operation on computer 1, the corresponding data display on the interactive interface can be obtained. Meanwhile, the visual flight demonstration can be realized on computer 2. The internal operation diagram of the simulation platform is shown in Fig. 11.

The 3D scene dynamic simulation results are shown in Fig. 12, where the blue hemisphere represents the range of the threat source and the red curves represent the trajectories of UAVs and their color fades after a certain period of time. Based on the visual simulation platform, we can see that this system shows the dynamic situation of multi-UAV search-attack mission planning as 3D animation, which makes the simulation results more exhaustive and makes it easier for the user to observe the process of mission execution.

VI. CONCLUSION

In this article, a dynamic discrete PIO algorithm based on hybrid architecture has been proposed to solve a search-attack mission planning problem for multiple UAVs. The dynamic concept, discretization processing, and solution acceptance strategy are adopted in the PIO algorithm to achieve excellent performance. Then, based on the RTSM and a probability map which is constructed and updated by Bayes formula, the objective function is designed in the D^2PIO algorithm for a cooperative distributed search-attack mission planning problem. The numerical simulation results and analysis have proved the effectiveness of the proposed method. Finally, the visual simulation platform accomplishes the dynamic simulation process of multi-UAV flight and task execution in the form of 3D animation, which is combined display software and HMI, utilizing Unity 3D and MFC.

In the future, we will improve the function of HMI and focus on cooperative mission planning for large-scale UAVs swarms with the situation assessment section to realize the autonomous decision-making based on the interactive 3D visual simulation platform.

REFERENCES

[1] L. Gupta, R. Jain, and G. Vaszkun
Survey of important issues in UAV communication networks
IEEE Commun. Surveys Tut., vol. 18, no. 2, pp. 1123–1152, Nov. 2015.

[2] T. H. Chung, G. A. Hollinger, and V. Isler
Search and pursuit-evasion in mobile robotics
Auton. Robots, vol. 31, no. 4, pp. 299–316, 2011.

[3] P. Lanillos, E. Besada-Portas, J. A. Lopez-Orozco, and J. M. de la Cruz
Minimum time search in uncertain dynamic domains with complex sensorial platforms
Sensors, vol. 14, no. 8, pp. 14131–14179, 2014.

[4] S. Perez-Carabaza, E. Besada-Portas, J. A. Lopez-Orozco, and J. M. de la Cruz
Ant colony optimization for multi-UAV minimum time search in uncertain domains
Appl. Soft Comput., vol. 62, pp. 789–806, 2018.

[5] P. Li and H. B. Duan
A potential game approach to multiple UAV cooperative search and surveillance
Aerosp. Sci. Technol., vol. 68, pp. 403–415, 2017.

[6] C. Ramirez-Atencia, G. Bello-Orgaz, M. D. R-Moreno, and D. Camacho
Solving complex multi-UAV mission planning problems using multi-objective genetic algorithms
Soft Comput., vol. 21, no. 17, pp. 4883–4900, 2017.

[7] S. S. Ponda, L. B. Johnson, A. Geramifard, and J. P. How
Cooperative mission planning for multi-UAV teams
in *Handbook of Unmanned Aerial Vehicles*. Dordrecht, Netherlands: Springer-Verlag, 2015, pp. 1447–1490.

[8] S. Moon, D. H. Shim, and E. Oh
Cooperative task assignment and path planning for multiple UAVs
in *Handbook of Unmanned Aerial Vehicles*, Dordrecht, Netherlands: Springer, 2015, pp. 1547–1576.

[9] S. Martinez, J. Cortes, and F. Bullo
Motion coordination with distributed information
IEEE Control Syst. Mag., vol. 27, no. 4, pp. pp. 75–88, Aug. 2007.

[10] S. Kar and J. M. F. Moura
Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise
IEEE Trans. Signal Process., vol. 57, no. 1, pp. 355–369, Jan. 2008.

[11] L. Ambroziak and Z. Gosiewski
Two stage switching control for autonomous formation flight of unmanned aerial vehicles
Aerosp. Sci. Technol., vol. 46, pp. 221–226, 2015.

[12] M. M. Polycarpou, Y. Yang, and K. M. A Passino
Cooperative search framework for distributed agents
in *Proc. IEEE Int. Symp. Intell. Control*, 2002, pp. 1–6.

[13] E. Edison and T. Shima
Integrated task assignment and path optimization for cooperating uninhabited aerial vehicles using genetic algorithms
Comput. Oper. Res., vol. 38, no. 1, pp. 340–356, 2011.

[14] T. Shima, S. J. Rasmussen, and A. G. Sparks
Multiple task assignments for cooperating uninhabited aerial vehicles using genetic algorithms
Comput. Oper. Res., vol. 33, no. 11, pp. 3252–3269, 2006.

[15] H. L. Choi, L. Brunet, and J. P. How
Consensus-based decentralized auctions for robust task allocation
IEEE Trans. Robot., vol. 25, no. 4, pp. 912–926, Aug. 2009.

[16] G. Oh, Y. Kim, and J. Ahn
Market-based distributed task assignment for rendezvous mission over networks with limited connectivity
IFAC Proc. Vol., vol. 47, no. 3, pp. 888–893, 2014.

[17] R. R. Pitre, X. R. Li, and R. Delbalzo
UAV route planning for joint search and track missions—An information-value approach
IEEE Trans. Aerosp. Electron. Syst., vol. 48, no. 3, pp. 2551–2565, Jul. 2012.

[18] D. Angley, B. Ristic, W. Moran, and B. Himed
Search for targets in a risky environment using multi-objective optimisation
IET Radar, Sonar Navigat., vol. 13, no. 1, pp. 123–127, 2018.

[19] Y. Chen, D. Yang, and J. Yu
Multi-UAV task assignment with parameter and time-sensitive uncertainties using modified two-part wolf pack search algorithm
IEEE Trans. Aerosp. Electron. Syst., vol. 54, no. 6, pp. 2853–2872, Dec. 2018.

[20] Z. Zhou *et al.*
When mobile crowd sensing meets UAV: Energy-efficient task assignment and route planning
IEEE Trans. Commun., vol. 66, no. 11, pp. 5526–5538, Nov. 2018.

[21] X. Hu, H. Ma, Q. Ye, and H. Luo
Hierarchical method of task assignment for multiple cooperating UAV teams
J. Syst. Eng. Electron., vol. 26, no. 5, pp. 1000–1009, 2015.

[22] M. H. Kim, H. Baik, and S. Lee
Response threshold model based UAV search planning and task allocation
J. Intell. Robot. Syst., vol. 75, no. 3-4, pp. 625–640, 2014.

[23] C. Gao, Z. Zhen, and H. Gong
A self-organized search and attack algorithm for multiple unmanned aerial vehicles
Aerosp. Sci. Technol., vol. 54, pp. 229–240, 2016.

[24] Z. Zhen, D. Xing, and C. Gao
Cooperative search-attack mission planning for multi-UAV based on intelligent self-organized algorithm
Aerosp. Sci. Technol., vol. 76, pp. 402–411, 2018.

[25] M. Sagliano
Pseudospectral convex optimization for powered descent and landing
J. Guid. Control Dyn., vol. 41, no. 2, pp. 320–334, 2017.

[26] H. Yang, X. Bai, and H. Baoyin
Rapid generation of time-optimal trajectories for asteroid landing via convex optimization
J. Guid. Control Dyn., vol. 40, no. 3, pp. 628–641, 2017.

[27] R. Chai, A. Savvaris, and A. Tsourdos
Trajectory optimization of space maneuver vehicle using a hybrid optimal control solver
IEEE Trans. Cybern., vol. 49, no. 2, pp. 467–480, Feb. 2017.

[28] R. Chai, A. Savvaris, A. Tsourdos, S. Chai, and Y. Q. Xia
Trajectory optimization of space maneuver vehicle using a hybrid optimal control solver
J. Guid. Control Dyn., vol. 40, no. 8, pp. 2091–2100, 2017.

[29] A. Asma and B. Sadok
PSO-based dynamic distributed algorithm for automatic task clustering in a robotic swarm
Procedia Comput. Sci., vol. 159, pp. 1103–1112, 2019.

[30] H. B. Duan, P. Li, and Y. H. Shi
Interactive learning environment for bio-inspired optimization algorithms for UAV path planning
IEEE Trans. Educ., vol. 58, no. 4, pp. 276–281, Nov. 2015.

[31] H. B. Duan and P. X. Qiao
Pigeon-inspired optimization: A new swarm intelligence optimizer for air robot path planning
Int. J. Intell. Comput. Cybern., vol. 7, no. 1, pp. 24–37, 2014.

[32] H. X. Qiu, H. B. Duan, and Y. M. Fan
Unmanned aerial vehicle close formation cooperative control based on predatory escaping pigeon-inspired optimization
Scientia Sinica Technologica, vol. 45, no. 6, pp. 559–572, 2015.

[33] H. B. Duan, Y. X. Yu, and Z. Y. Zhao
Advancements in pigeon-inspired optimization and its variants
Sci. China Inf. Sci., vol. 62, no. 7, 2019, Art. no. 070201.

[34] L. Chen, H. B. Duan, Y. M. Fan, and C. Wei
Multi-objective clustering analysis via combinatorial pigeon inspired optimization
Sci. China Technological Sci., vol. 63, no. 7, pp. 1302–1313, 2020.

[35] H. B. Duan, M. Z Huo, and Y. H. Shi
Limit-cycle-based mutant multi-objective pigeon-inspired optimization
IEEE Trans. Evol. Comput., vol. 24, no. 5, pp. 948–959, Oct. 2020.

[36] W. Y. Ruan and H. B. Duan
Multi-UAV obstacle avoidance control via multi-objective social learning pigeon-inspired optimization
Frontier Inf. Technol. Electron. Eng., vol. 21, no. 5, pp. 740–748, 2020.

[37] A. Salman, I. Ahmad, and S. Al-Madani
Particle swarm optimization for task assignment problem
Microprocessors Microsyst., vol. 26, no. 8, pp. 363–371, 2002.

[38] B. Pang, Y. Song, C. Zhang, H. Wang, and R. Yang
Autonomous task allocation in a swarm of foraging robots: An approach based on response threshold sigmoid model
Int. J. Control, Autom. Syst., vol. 17, no. 4, pp. 1031–1040, 2019.

[39] E. P. Anderson, R. W. Beard, and T. W. Mclain
Real-time dynamic trajectory smoothing for unmanned air vehicles
IEEE Trans. Control Syst. Technol., vol. 13, no. 3, pp. 471–477, May 2005.

[40] M. Elbanhawi, M. Simic, and R. N. Jazar
Continuous path smoothing for car-like robots using B-spline curves
J. Intell. Robotic Syst., vol. 80, no. 1, pp. 23–56, 2015.

[41] D. Tian and Z. M. Shi
Modified particle swarm optimization and its applications
Swarm Evol. Computation, vol. 41, pp. 49–68, 2018.

Haibin Duan (Senior Member, IEEE) received the Ph.D. degree in control theory and engineering from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2005.

He was a Senior Visiting Scholar of The University of Suwon, Hwaseong, South Korea, in 2011. He is currently a Full Professor with the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China, where he is the Vice Director of the State Key Laboratory of Virtual Reality Technology and Systems and the Head of the Bio-Inspired Autonomous Flight Systems Research Group. He is also a Visiting Research Fellow with Peng Cheng Laboratory, Shenzhen, China. He has authored or coauthored more than 70 publications. His current research interests are bio-inspired intelligence, and multi-UAV autonomous swarm control.

Prof. Duan was the recipient of the National Science Fund for Distinguished Young Scholars of China in 2014. He is also enrolled in the Chang Jiang Scholars Program of China, Scientific and Technological Innovation Leading Talent of "Ten Thousand Plan"-National High Level Talents Special Support Plan, and Top-Notch Young Talents Program of China.

Jianxia Zhao received the B.S. degree in automation science from Beihang University, Beijing, China, in 2018. She is currently working toward the Master's degree at the BAFS Research Group, State Key Laboratory of Virtual Reality Technology and Systems, School of Automation Science and Electrical Engineering, Beihang University.

Her current research interests include unmanned aerial vehicle cooperative control and mission planning.

Yimin Deng received the B.S. and Ph.D. degrees from the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China, in 2011 and 2017, respectively.

He is currently an Assistant Professor with the School of Automation Science and Electrical Engineering, Beihang University. His current research interests include biological computer vision and autonomous flight control.

Dr. Deng is enrolled in the Young Elite Scientists Sponsorship Program by Chinese Association for Science and Technology and Young Top Talent Support Program by Beihang University.

Yuhui Shi (Fellow, IEEE) received the Ph.D. degree in electrical and electronic engineering from Southeast University, Nanjing, China, in 1992.

From January 2008 to August 2017, he was with the Department of Electrical and Electronic Engineering, Xi'an Jiaotong-Liverpool University, Suzhou, China, from October 1998 to December 2007 with the Electronic Data Systems Corporation, Indianapolis, IN, USA, and from October 1995 to September 1998, with the Purdue School of Engineering and Technology, Indianapolis. He is a Chair Professor with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China. His main research interests are swarm intelligence and applications.

Dr. Shi is the Editor-in-Chief of the *International Journal of Swarm Intelligence Research*, and an Associate Editor for the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION.

Xilun Ding received the B.Eng. degree in mechanical engineering from the Zhengzhou Institute of Technology, Zhengzhou, China, in 1991, and the M.Eng. degree in mechanical design and Ph.D. degree in mechatronics and automation from the Harbin Institute of Technology, Harbin, China, in 1994 and 1997, respectively.

From 1997 to 1998, he was a Postdoctoral Research Fellow with Robotics Institute, Beihang University, Beijing, China, where he became an Associate Professor in 1998, and has been a Professor since 2004. His research interests include aerial robots, bioinspired robots, and deployable mechanisms.

Dr. Ding currently serves as an Associate Editor of *the Journal of Mechanisms and Robotics-Transactions* of the ASME. He received the National Science Fund for Distinguished Young Scholars of China Award in 2011.