Hindawi
Complexity
Volume 2022, Article ID 9219805, 14 pages
https://doi.org/10.1155/2022/9219805

Research Article

Modeling and Solving the Dynamic Task Allocation Problem of Heterogeneous UAV Swarm in Unknown Environment

Qiang Peng,[1] Husheng Wu ,[1] and Na Li[2]

[1] *College of Equipment Support and Management, Engineering University of PAP, Xi'an 710086, China*
[2] *Foundation Department, Engineering University of PAP, Xi'an 710086, China*

Correspondence should be addressed to Husheng Wu; wuhusheng0421@163.com

Received 12 January 2022; Accepted 15 March 2022; Published 22 April 2022

Academic Editor: Xianming Zhang

As a NP-hard problem that needs to be solved in real time, the dynamic task allocation problem of unmanned aerial vehicle (UAV) swarm has gradually become a difficulty and hotspot in the current planning field. Aiming at the problems of poor real-time performance and low quality of the solution in the dynamic task allocation of heterogeneous UAV swarm in uncertain environment, this paper establishes a dynamic task allocation model that can meet the actual needs and uses the binary wolf pack algorithm (BWPA) to solve it, so as to propose a dynamic task allocation method of heterogeneous UAV swarm in uncertain environment. In this method, a dynamic mechanism of attacking while searching and priority attacking of important targets is designed. A dynamic task allocation model of multitarget, multitask, heterogeneous multiaircraft platform and multiconstraint is established based on the target cost-effectiveness ratio and task execution time window. In addition, one-dimensional 0–1 coding method is adopted to encode the task allocation scheme. Furthermore, the wolf pack algorithm (WPA) is introduced in brief. This paper focuses on the BWPA with the good computational robustness and strong global search ability to solve the dynamic allocation model. According to the simulation results, the designed task allocation method not only has good adaptability to the change of target and UAV number, as well as good stability and scalability, but also can effectively solve the dynamic task allocation problem of heterogeneous UAV swarm in unknown environment. Therefore, the established model and solution method can provide a useful reference for task allocation and other related problems.

1. Introduction

With better application prospects in the military field, unmanned aerial vehicle (UAV) performs reconnaissance, surveillance, target acquisition, real-time strike, and other tasks. However, it is difficult for a single UAV to meet the needs of increasingly complex tasks. Thus, the application of UAV swarm becomes more and more important, and research topic of UAV swarm task allocation is carried out. The task allocation of the UAV swarm is to conduct effective task allocation and determine the task plan of each UAV according to the task requirements, UAV characteristics, and the mission load performance, so as to give full play to the role of each UAV and improve the overall efficiency of the swarm. In line with the certainty of allocation environment, UAV swarm task allocation can be divided into two types of task allocation: "static and dynamic" [1, 2]. The swarm status in static task allocation is determined and the location of the target is known and fixed, so that the allocation is relatively simple. However, in dynamic task allocation, the target location and environments are constantly changing. Therefore, UAV swarm dynamic task allocation is not only a NP-hard problem that needs to be solved quickly [3], but also a key, difficult, and hot issue in the current task planning field [4]. At present, the research on UAV dynamic task allocation at home and abroad mainly focuses on the establishment and solution of the model.

There are a lot of research results on dynamic task allocation model. For example, Shima T [5] proposed a cooperative multiple task assignment problem (CMTAP) model for UAV. Based on [5], paper [6] improved the CMTAP model and established an ex-tended cooperative multitask assignment model, but the constraints of the model were still not fully considered. In [7], the dynamic cooperative attack process was simplified into a two-stage static WTA process, and a dynamic WTA model was proposed to better describe the complex cooperative attack problem. In addition, [8] transformed dynamic task allocation into static task allocation in multiple stages and constructed a multi-UAV dynamic reconnaissance resource allocation model, which could better improve the overall efficiency of multi-UAV performing dynamic reconnaissance tasks. Meanwhile, [9] constructed a state information description model for unmanned aerial vehicles to perform tasks, which reduced the scale, time, and communication complexity of task allocation. When adding new tasks or platform loss, [10] adopted the way of dynamic task local adjustment to build a multibase and multi-UCAV task allocation model, which improved the task allocation efficiency and enhanced the stability of the platform. Besides, [11] built up an alliance formation model but ignored the impact of communication constraints such as communication distance and communication delay. Additionally, [12] constructed a multi-UAV task allocation model based on interval information environment according to uncertain information about revenue damage cost index, target value, and range cost index.

There are many methods to solve the dynamic task allocation problem. Currently, the commonly used methods mainly include market-based algorithm and intelligent optimization algorithm. Reference [13] studied the cooperative search and simultaneous attack of UAV group on multiple targets and proposed a task allocation algorithm that can obtain the maximum overall efficiency in real-time environment. Moreover, [14] proposed a contract network algorithm based on concurrency. The algorithm can not only quickly and effectively deal with various emergencies in task execution, but also better meet the needs of dynamic task allocation. For the timing problem of task allocation, [15] proposed a genetic algorithm based on matrix coding, which can better improve the efficiency of task allocation. At the same time, [16] used the improved k-means swarming algorithm to swarm and group the "UAV task" set and then used the split particle swarm optimization algorithm to reallocate the tasks of multiple UAVs in the group. Although this strategy can obtain higher returns at the cost of reducing time, it is difficult to obtain the global optimal solution. Beyond that, [17] proposed a variable structure discrete dynamic Bayesian network reasoning algorithm for UAV task allocation in uncertain environment. Reference [18] uses dynamic game theory to decompose the dynamic task allocation problem in swarm air combat into a game problem in a single UAV and proposed a particle swarm optimization algorithm for predator and prey. It is noteworthy that this method is generally only suitable for the operation between small-scale swarms, but it is difficult to deal with the confrontation between large-scale swarms. Meanwhile, [19] combined the advantages and disadvantages of particle swarm optimization algorithm and bacterial foraging algorithm and proposed a hybrid bacterial foraging algorithm to solve the dynamic task allocation problem of multiple UCAVs.

To sum up, although a large number of research achievements have been made in the establishment and solution of the model, the following problems should still be solved: (1) The existing UAV dynamic task allocation models are ideal, easy to solve, and poorly dynamic. (2) There is little research on the fully dynamic task allocation solution strategy. Although the current dynamic task allocation adjusts the tasks on the preallocation scheme obtained in line with some target information, it is difficult to obtain information in advance in the actual task. (3) For the dynamic adjustment of large-scale UAV swarm, the solution space increases, the solution set increases exponentially, and the effectiveness of the algorithm decreases obviously. As a result, it is difficult to ensure the real-time and accurate allocation.

In order to get closer to the combat reality, a fully dynamic task allocation strategy has been adopted, and a dynamic task allocation method of UAV swarm based on BWPA is proposed. This method not only establishes a dynamic allocation model of attacking while searching and attacking on important targets first, but also uses the BWPA [20] with good computational robustness and strong global search ability to solve the model. As shown by the simulation results this method can better meet the needs of dynamic task allocation and can also provide a useful reference for solving problems in related systems such as relay networks and D2D systems [21, 22].

The structure of the paper is as follows: In Section 1, the motivation and current research status are discussed. In Section 2, a dynamic task allocation model is established. In Section 3, the novel dynamic task allocation mechanism is proposed. In Section 4, the solution method of discrete BWPA is introduced. In Section 5, the experiments are performed to evaluate the effectiveness and dynamic adaptability of the proposed dynamic task allocation method. In Section 6, the conclusion is drawn and the future work is discussed.

2. The Model of Dynamic Task Allocation

2.1. Problem Description

2.1.1. Task Allocation Scenarios and Constraints. The task allocation scenario set in this paper is UAV swarm cooperative operation; namely, in the battlefield, a heterogeneous swarm composed of search UAV (S-UAV) in number of NS and attacking UAV (A-UAV) in number of NA performs "reconnaissance attack" tasks on the unknown targets in number of NT to be attacked.

In the scenario of UAV task allocation, the dynamic resource allocation for static targets is mainly studied. In order to simplify the problem and build the model, the influence of communication design and data timeliness [23–25] is not considered temporarily, and the following restrictions for UAV swarm operation are set:

(1) The battlefield environment is a two-dimensional plane, and there are no forbidden fly zones or various obstacles.

(2) There is no gray or black battlefield information, and the communication between UAVs works smoothly.

(3) The UAV is regarded as a particle, and the flight altitude and turning radius of the UAV are not considered temporarily.

(4) Assuming that the UAV reaches the target location, it will complete the reconnaissance or attack mission.

(5) The efficiency gain is not considered temporarily when UAVs perform tasks cooperatively.

(6) It is assumed that enemy targets are independent fixed targets, and the effect of targets on UAV is not considered temporarily.

2.1.2. The Description of Battlefield Environment. The battlefield environment is set as a bounded area $L_x \times L_y$. There are a series of static targets, the targets pose no threat to UAV, and the number and location of targets are unknown. As shown in Figure 1, the Cartesian grid recognition map is used to describe the battlefield environment, and the battlefield environment is discretized into units of number $n_x \times n_y$. Besides, there is only one target in each unit at most.

2.1.3. Battlefield Subject Description. UAV swarm and target to be attacked are the two main bodies of mutual confrontation in the swarm battlefield. Their respective attributes are described as follows.

In the UAV set $U^k = \left\{ U^k_{N_k} | k = 1, 2, \ldots, N_U \right\}$, N_k represents the total number of the k UAVs, k represents the type, and N_U represents the total number in types, and $NV = N_U \times N_k$ represents the total number. This paper takes $N_U = 2$, which represents two types, respectively: reconnaissance UAV and attack UAV, reconnaissance UAV in number of NS and attack UAV in number of NA. UAV attributes are represented by four tuples $\langle ID_U, Loc_U, Ru, S_U \rangle$, and the meaning of each symbol is shown in Table 1. Notably, Ru is the maximum resource carried by the UAV at a time, and the corresponding UAV can attack targets requiring Ru unit resources at most.

In the target set $T = \{T_i | i = 1, 2, \ldots, N_T\}$, N_T represents the total number of targets. The attribute of the target is represented by four tuples $\langle ID_T, Loc_T, \omega_T, RT \rangle$ (see the meaning of each symbol in Table 1).

2.2. Mathematical Model. In UAV swarm operation, given the overall benefits of the swarm and learning reference from common modeling methods [13], the UAV swarm task allocation model can be constructed as follows.

Assuming that the net income of UAV performing each task meets additivity, the task decision variables $x_{i,j}$ can be defined as follows:

$$x_{i,j} = \begin{cases} 1, & Target\ j \text{ assigned to } UAV\ i, \\ 0, & \text{Others.} \end{cases} \tag{1}$$

In this equation, $i = 1, 2, \cdots, N_A$, $j = 1, 2, \cdots, N_T$.

Then, the mathematical model of task allocation of UAV swarm can be expressed as

$$\max f = \sum_{i=1}^{N_A} \sum_{j=1}^{N_T} S_{i,j} \cdot x_{i,j}. \tag{2}$$

In the above equation, $S_{i,j}$ denotes the flight range of the UAV i to perform the task on the target j.

The constraints are as follows:

(1) Each task for each target can only be completed once by one UAV:

$$\sum_{i=1}^{N_A} x_{i,j} = 1, \quad j = 1, 2, \ldots, N_T. \tag{3}$$

(2) The resources carried by the UAV performing the task on each target must not be less than that of the required:

$$\sum_{i=1}^{N_A} Ru_i \cdot x_{i,j} \geq RT_j, \quad j = 1, 2, \ldots, N_T. \tag{4}$$

(3) The total amount of resources consumed by UAV i in performing tasks must not exceed its own resources:

$$\sum_{j=1}^{N_T} Ru_i \cdot x_{i,j} \leq Ru_i, \quad i = 1, 2, \ldots, N_A. \tag{5}$$

(4) The total range of each UAV during mission flight shall not exceed its maximum range:

$$\sum_{j=1}^{N_T} S_{i,j} \cdot x_{i,j} + d_i \leq \mathrm{Smax}_i, \quad i = 1, 2, \ldots, N_A. \tag{6}$$

In the above equation, d_i represents the return voyage of UAV i, and the $S\mathrm{max}_i$ represents the maximum voyage of UAV i.

3. Dynamic Allocation Mechanism

In order to get closer to the combat reality and meet the needs of dynamic tasks, this paper adopts the mode of searching and attacking. UAV swarm completely relies on reconnaissance UAVs to obtain battlefield target

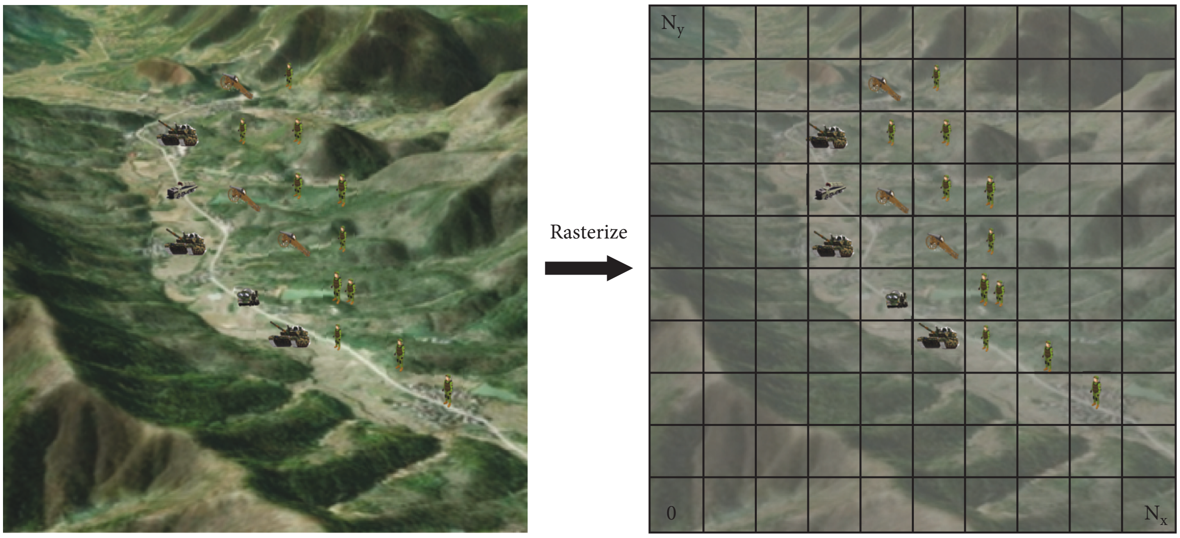


Figure 1: Schematic diagram of environment rasterization.

Table 1: Symbolic description.

Type	Symbol	Significance
Attribute symbol of UAV	ID_U	Identification number of UAV
	Loc_U	Position coordinates of UAV
	Ru	Attack resources carried of UAV
	S_U	Maximum driving range of UAV
Attribute symbol of the target	ID_T	Identification number of the target
	Loc_T	Location coordinates of the target
	ω_T	Value of attack target
	RT	Required attack resources

information. Before fully searching the battlefield, the information of UAV swarm on battlefield targets always keeps growing. This requires the UAV swarm to make full use of the known information, so that the allocation decision can be favorable to the overall effectiveness of the swarm. As displayed in Figure 2, multiple reconnaissance UAVs search the target according to the search cycle, so as to obtain the situation information in the search cycle. The swarm system generates attack sequences according to certain rules, calls the task allocation module for task allocation, and then transmits the allocation results to the attack UAV. Apart from that, each attack UAV parses the attack command and attacks the corresponding target. If no target is detected in the search cycle and there is no previous unexecuted target, the task allocation module will not be called in this cycle.

3.1. Multimachine Slice and Area Search. As shown in Figure 3, the reconnaissance UAV divides the battlefield, searches the battlefield according to the reciprocating search method in document [26], and quickly finds the target. In order to reduce the number of turns, the UAV is designed to fly along the long side of the rectangle.

3.2. Dynamic Distribution Mechanism. In this paper, only the attack task of the target is allocated, so that the target is directly used to represent the task. In order to solve the prominent problems such as slow computing speed and poor dynamics of centralized task allocation method, a dynamic allocation mechanism is designed. To be specific, it mainly realizes the following functions: (1) selecting the appropriate task allocation time to avoid the waste of computing resources caused by frequent allocation; (2) achieving the optimization of the objective function according to the attack sequence generated by the efficiency cost ratio of each target; (3) determining the target scale of task allocation with an appropriate task window to improve the real-time performance of task allocation.

3.2.1. Allocation Timing. According to the task completion of the target in the operation process, the target is divided into three categories: the target attacked, target to be executed, and target to be assigned. The task allocation of UAV swarm is driven by time and events together, and the search process is classified into several search cycles TS. When reconnaissance UAV searches every cycle TS, it will judge whether to allocate tasks or not. If there are new targets or targets missed in the previous cycle, it will record them as target task set TR to be executed and then allocate it. Otherwise, there is no need to allocate. In order to ensure that each search cycle TS reaches any position in the battlefield to complete the attack task against the

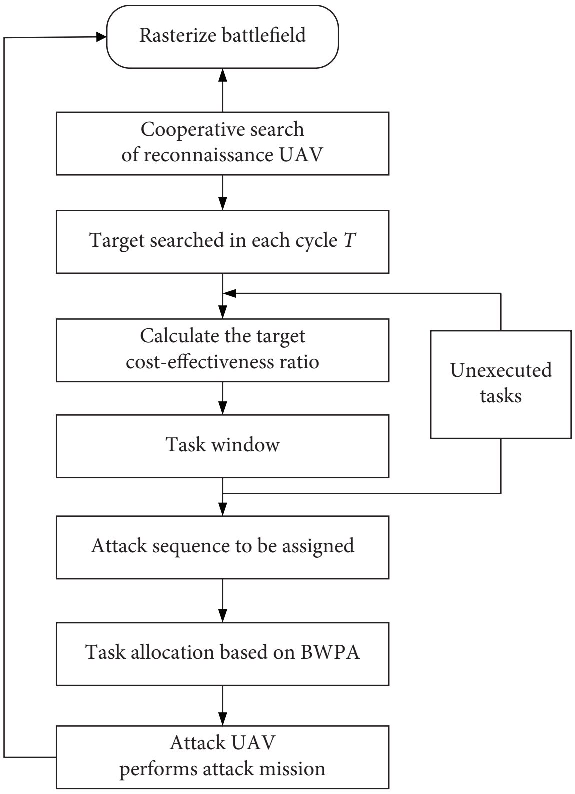

Rasterize battlefield
Cooperative search of reconnaissance UAV
Target searched in each cycle T
Calculate the target cost-effectiveness ratio
Unexecuted tasks
Task window
Attack sequence to be assigned
Task allocation based on BWPA
Attack UAV performs attack mission

Figure 2: Flowchart of UAV swarm executing dynamic tasks.

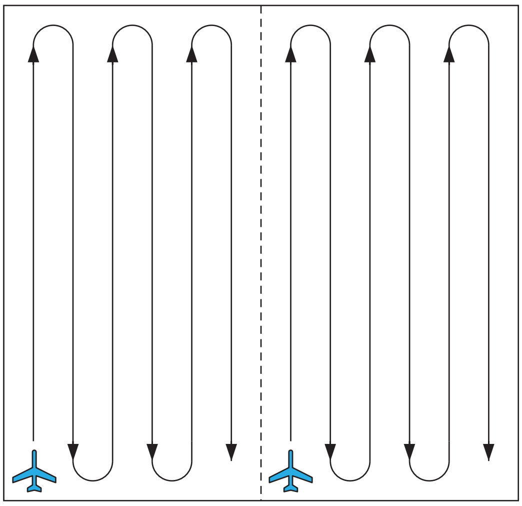


Figure 3: Multislice coverage search mode.

target allocated in the previous cycle, the value of search cycle TS should be greater than the time required for the UAV to fly the longest path, as displayed in (7).

$$TS \geq \frac{D\max}{v}, \tag{7}$$

where $D\max$ represents the longest path among any two points on the battlefield; v represents the flight speed of UAV.

3.2.2. Attack Sequence. Similar to the method in [27], most of the current dynamic task allocation methods determine the order of target execution according to the time of occurrence or the distance. This method is clear-cut. However, in the actual battlefield, the target often has certain timeliness, and the target with high value needs to be attacked first. Therefore, this paper adopts the method of “attacking while discovering, attacking first with high value” and takes the attack efficiency cost ratio as the ranking basis of target attack order. In the current search cycle, the closer it is to the swarm, the greater the value and the less resource required, and the more the priority attacks are needed. The attack cost-effectiveness ratio L_j of the target T_j can be defined as

$$L_j = \frac{\sum_{i=1}^{N_A} Ru_i(t)}{d\min_{ij}(t)} \cdot \frac{\omega_j}{RT_j}. \tag{8}$$

In (8), the target number is $j = 1, 2, ..., N_T$, $\sum_{i=1}^{N_A} Ru_i(t)$ represents the sum of all resources of the UAV swarm at time t, $d\min_{ij}(t)$ represents the nearest distance from the UAV swarm at time t, ω_j represents the value of the target j, and RT_j represents the resources required for the target j to be destroyed.

After the current targets to be executed are arranged in descending order according to the attack cost-effectiveness ratio, the attack sequence is

$$T' = \left\{T_1', T_2', \ldots, T'_{N_t} | L_j \geq L_{j+1}, j = 1, 2, \ldots, N_t - 1\right\}, \tag{9}$$

where N_t denotes the total number of targets to be executed at the current time.

3.2.3. Task Window. For the task allocation problem with timeliness requirements, the allocation method is required to quickly respond to the newly discovered targets and generate an allocation scheme, and the implementation of the scheme can maximize the overall benefit of UAV swarm as much as possible. As shown in (10), in order to reduce the number of optimization targets, shorten the calculation time, and improve the allocation efficiency, the task window with length M is used to select the first target in the attack sequence for the priority attack. When the number of targets to be attacked is less than the window length M, the task to be assigned in the window is target N_t to be attacked. When the number of targets to be attacked is greater than the window length M, the tasks to be assigned in the window are the first targets M in the attack sequence.

$$T_{Win} = \{T_1', T_2', \ldots, T_k' | j = 1, 2, \ldots, k\},$$
$$k = \min\{N_t, M\}. \tag{10}$$

To conclude, when the reconnaissance UAV finds the target, it will not be allocated immediately, but it will be done only after the search cycle TS, which will cause few targets or waste of computing resources due to frequent allocation.

4. Solution of Task Allocation Problem Based on BWPA

4.1. Task Allocation Scheme Code. The decision variables $x_{i,j}$ of the model consist of two dimensions. The task allocation scheme formed is difficult to express intuitively, and it is not convenient to adopt intelligent optimization algorithm for scheme optimization. As shown in Figure 4, a new one-dimensional binary code X is designed to represent the task allocation scheme.

In Figure 4, Y represents the target vector determined by the combat target task table, with a total of N_T targets. At the same time, the allocation scheme vector corresponding to each target is X, and the subscript j of X represents the target number, $j = 1, 2, \ldots, N_T$, and X_j represents the allocation scheme of the target j. If the element is $x_{i,j} = 1$, it indicates that the swarm UAV_i performs an attack task on the target j; if the element is $x_{i,j} = 0$, it reveals that the swarm UAV_i does not attack the target j.

In order to intuitively understand the above coding method, an example is introduced for explanation. Three heterogeneous swarms of different UAVs are set to attack four targets. Then, it can be obtained that $i = 1, 2, 3$ and $j = 1, 2, 3, 4$ in the decision variable $x_{i,j}$. As shown in Figure 5, the target number vector is Y, if the one-dimensional binary code of the allocation scheme is $\mathbf{X} = (0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0)$; namely, the UAVs numbered 3, 2, 3, 1 perform attack tasks on the targets numbered 1, 2, 3, 4, respectively. Therefore, the corresponding decision variables are $x_{3,1} = 1$, $x_{2,2} = 1$, $x_{3,3} = 1$, $x_{1,4} = 1$.

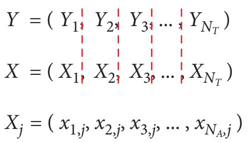

$Y = (Y_1, Y_2, Y_3, \ldots, Y_{N_T})$
$X = (X_1, X_2, X_3, \ldots, X_{N_T})$
$X_j = (x_{1,j}, x_{2,j}, x_{3,j}, \ldots, x_{N_A,j})$

Figure 4: Schematic diagram of distribution scheme.

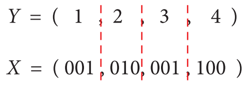


Figure 5: Coding diagram.

Thus, one-dimensional binary code X can be used to represent the meaning and allocation scheme represented by two-dimensional decision variables $x_{i,j}$. This is convenient not only for visual representation, but also for algorithm optimization.

4.2. Binary Wolf Pack Algorithm. Wolf pack algorithm (WPA) [28] is a swarm intelligence optimization algorithm proposed by Hu Sheng et al. based on the two rules that "the winner is the king and the strong survives," which further analyzes the three behaviors of "scout wolf wandering, lead wolf calling, and ferocious wolf sieging" in wolf hunting. Compared with genetic algorithm and particle swarm optimization algorithm, WPA has the advantages of small parameter sensitivity, strong robustness, fast convergence speed, and high solution accuracy. Because of its better performance, it has become an effective algorithm for solving NP-hard problems and has been widely applied in many fields [29–34]. In order to apply WPA to discrete space, binary coding is introduced into WPA, and the relevant definitions are modified to obtain BWPA [20].

4.2.1. Relevant Definitions. Let the solution space be an Euclidean space $m \times N$, the location of wolf i is $X_i = (x_{i1}, x_{i2}, \ldots, x_{ij}, \ldots, x_{im})$, $i = 1, 2, \ldots, N$, $j = 1, 2, \ldots, m$, N is the number of populations, and m is the coding length. Due to binary encoding, $x_{i,j}$ can only take 0 or 1. The odor concentration of prey where the wolf is located is the fitness value $Y = f(X)$ of the objective function. As shown in (11), the distance between wolf p and wolf q is represented by binary coded Manhattan distance.

$$L(p, q) = \sum_{j=1}^{m} \left| x_{pj} - x_{qj} \right|,$$
$$p, q \in \{1, 2, \ldots, N\}. \tag{11}$$

Definition 1. Reverse: to reverse $x_{i,j}$ is to assign a value to $X_i = (x_{i1}, x_{i2}, \ldots, x_{ij}, \ldots, x_{im})$ in the position of wolf i according to (12).

$$x_{ij} = \begin{cases} 1, & \text{if } x_{ij} = 0, \\ 0, & \text{if } x_{ij} = 1. \end{cases} \tag{12}$$

Definition 2. The kinematic operator: let the position of wolf i be $X_i = (x_{i1}, x_{i2}, \ldots, x_{im})$; M represents the set of coding bits that can be inverted and is not an empty set, namely, the movable range of wolf; r represents the number of coding bits to be inverted, namely the motion step of the wolf. The kinematic operator $\Theta(X_i, M, r)$ indicates that r coding bits are randomly selected in M for inversion.

Example 1. If $X_i = (1,0,1,1,0,0)$, $M = \{1,3\}$, and $r = 1$, then $\Theta(X_i, M, r) = (0,0,1,1,0,0)$ or $(1,0,0,1,0,0)$.

4.2.2. Description of Intelligent Behaviors and Rules. The BWPA also consists of the lead wolf generation rule "the winner is the king," "survival of the fittest" wolf pack update mechanism, and three intelligent behaviors of "wandering, summoning, and siege." The first two are consistent with the basic WPA [28], which will not be repeated here. The following part mainly describes the three intelligent behaviors.

(1) Wandering behavior. The wolf with the best fitness value is the lead wolf, and others are probe wolves. The fitness value Y_i of the current position of probe i is calculated and recorded. The probe wolf advances one step in h directions with the walking step step$_a$, namely, h times of motion operator $X_{ip}^{\text{new}} = \Theta(X_i, M_a, \text{step}_a)$, $M_a = \{1, 2, \ldots, m\}$, $p \in H$, and $H = \{1, 2, \ldots, h\}$ is executed for the probe wolf position X_i. At the same time, the fitness value corresponding to the position after each step is recorded, and then greedy decision is made. Let the corresponding fitness value obtained after moving in the p^{th} direction be Y_{ip}. If p^* meets the conditions in (13), one step in the direction p^* is advanced and the wolf detection position X_i is updated. The above walking behavior will be repeated until $Y_i > Y_{\text{lead}}$, or the number of walking T reaches the maximum number of walking $T_{\max}$, then turning to the calling behavior.

$$\begin{cases} p* \in \max\left\{Y_{ip}, p \in H\right\}, \\ Y_{ip*} > Y_{i0}. \end{cases} \tag{13}$$

In (13) $Y_{ip} = f(X_{ip}^{\text{new}})$; Y_{i0} represents the fitness value of the position where the wolf i is before migration; h represents a random integer between $[h_{\min}, h_{\max}]$.

(2) Summoning behavior. All wolves except the lead wolf are fierce wolves. The lead wolf initiates a call. The fierce wolf i quickly approaches the position X_d of the lead wolf with a large attack step step$_b$ according to the equation below:

$$X_i^{\text{new}} = \Theta(X_i, M_b, \text{step}_b). \tag{14}$$

M_b in (14) is obtained according to

$$M_b(k) = \begin{cases} j, k = k+1, j = j+1, & \text{if } x_{dj} \neq x_{ij}, \\ \text{null}, k = k, j = j+1, & \text{if } x_{dj} = x_{ij}. \end{cases} \tag{15}$$

In (15) $j = 1, 2, \ldots, m$, the initial value of k is 1, and null represents a null value. M represents the set of coding bits of different values at the corresponding positions of X_i and X_d.

When M is an empty set, the fierce wolf i rushes one step according to the random motion operator $\Theta(X_i, M_a, 1)$.

$Y_i = f(X_i^{\text{new}})$ is set as the fitness value of the location of fierce wolf i. If $Y_i > Y_{\text{lead}}$, then $Y_{\text{lead}} = Y_i$; fierce wolf i replaces the lead wolf and initiates a call. Otherwise, the attack continues until the conditions in (16) are met; then the call is end and turns into the siege.

$$\begin{cases} d_{is} < d_{\text{near}}, \\ d_{\text{near}} = \lceil \dfrac{m}{\omega} \rceil. \end{cases} \tag{16}$$

In the above equation, d_{is} represents the distance between the fierce wolf i and the lead wolf; d_{near} represents the determination distance and ω denotes the distance determination factor.

(3) Siege behavior. Taking the current position X_d of the wolf as the position of the prey, the fierce wolf sieges the prey according to

$$X_i^{\text{new}} = \Theta(X_i^*, M_a, \text{step}_c). \tag{17}$$

In the equation $X_i^* = \Theta(X_i, M_b, \text{step}_c)$, step$_c$ refers to the attack step size.

The wolf makes greedy decisions based on the size of the fitness values of the anterior and posterior positions.

The three steps, walking step step$_a$, rushing step step$_b$, and attacking step step$_c$, involved in the algorithm are integers, which represent the fineness of wolf search. There are usually the following relationships:

$$\lceil \frac{m}{10} \rceil \geq \text{step}_a \geq \text{step}_b \geq \text{step}_c = 1. \tag{18}$$

(4) Repair mechanism. The wolves R are eliminated according to the updated scale factor β, and then R new wolves are randomly generated. R represents a random integer within $[N/(2\beta), N/\beta]$. All wolves $X_i = (x_{i1}, x_{i2}, \ldots, x_{ij}, \ldots, x_{im})$ must meet relevant constraints; otherwise repeat the motion operator $X_i^{\text{new}} = \Theta(X_i, M_R, 1)$ for repair. The set M_R is obtained by

$$\lceil \frac{m}{10} \rceil \geq \text{step}_a \geq \text{step}_b \geq \text{step}_c = 1. \tag{19}$$

In the above equation, $j = 1, 2, \ldots, m$, the initial value of k is 1, and null represents a null value.

4.2.3. Algorithm Process. The specific steps of BWPA algorithm are as follows:

Step 1. Initialization: The number of wolf population is N, the initial position of the wolf is X_i, the maximum number of walks is $T_{\max}$, the maximum number of iterations is $k_{\max}$, the walk step is step$_a$, the attack step is step$_b$, and the attack step is step$_c$, the distance determination factor is ω, and the update scale factor is β.

Step 2. Selection of the lead wolf. The other wolves are probe ones and walk until $Y_i > Y_{\text{lead}}$ or $T > T_{\max}$ and turn to Step3.

Step 3. The fierce wolf runs towards the prey according to (14). If $Y_i > Y_{\text{lead}}$, then $Y_{\text{lead}} = Y_i$, and the fierce wolf i

replaces the lead wolf and initiates a call. Otherwise, attack continues until $d_{\mathrm{is}} < d_{\mathrm{near}}$; then switch to Step4.

Step 4. The fierce wolf sieges according to (17).

Step 5. Updating the wolf pack according to the wolf pack update mechanism and repairing the wolves that do not meet the conditions.

Step 6. Judging whether the termination conditions are met. If so, the optimal solution is output, that is, the wolf position X_d and its corresponding fitness value Y_{lead}. Otherwise, go back to Step2.

4.3. Time Complexity Analysis. Time complexity is an important embodiment of the efficiency of the algorithm. This paper uses the same method in [35, 36] to analyze the time complexity of the solution method of task allocation problem based on BWPA.

In WPA, set the population size as N and the individual dimension as m. When the step size is $step$, the update scale factor is β, the maximum number of walks is $T_{\max}$, and the determination distance d_{near} and other parameters have an initialization time of η_0, a random number generation time of η_1, and a model solution time of $f(m)$, the time complexity of the initialization phase is

$$o\left(\eta_0 + N\left(m\eta_1 + f(m)\right)\right) = o(m). \qquad (20)$$

According to the fitness value, the time for selecting the lead wolf is η_2, the time for detecting the wolf to execute the wandering strategy is η_3, the time for artificial wolf i to execute the calling behavior is η_4, the time required for artificial wolf i to approach the position of the lead wolf in each dimension is η_5, the time for judging whether to siege is η_6, and the time for siege is η_7; then the time complexity of this stage is

$$o\left(N\left(\eta_2 + \eta_3 + \eta_4 + n\eta_5 + \eta_6 + \eta_7\right)\right) = o(m + f(m)). \qquad (21)$$

The time complexity of each wolf swarm (WPA) is basically the optimal solution:

$$T_m = o(m) + o(m + f(m)) = o(m + f(m)). \qquad (22)$$

Analyze the process of the proposed method. In the initialization stage, the encoding time of the task allocation scheme is t_1, the decoding time of the task allocation scheme is t_2, and the other generation parameters, dimensions and model solutions are the same as those of WPA. Then the time complexity of the initialization phase of the proposed method is

$$o\left(\eta_0 + N\left(n\eta_1 + f(n) + t_1 + t_2\right)\right) = o(n). \qquad (23)$$

In addition, the selection of the lead wolf and the execution of intelligent behaviors such as wandering, summoning, and siege are the same as the WPA algorithm, so the time complexity of this stage is

$$o\left(N\left(\eta_2 + \eta_3 + \eta_4 + n\eta_5 + \eta_6 + \eta_7\right)\right) = o(\mathrm{n} + f(n)). \qquad (24)$$

Therefore, the total time complexity of each iteration of the proposed method is

$$o(n) + o(n + f(n)) = o(n + f(n)). \qquad (25)$$

To sum up, compared with WPA, the time complexity of the proposed method has not changed, and the operation efficiency of the algorithm has not been reduced.

5. Numerical Simulation and Analysis

The performance of the proposed dynamic task allocation method is verified by two sets of examples. The simulation environment is HONOR Magic Book Pro, Windows 10, Intel Core i5-10210U, and the program is implemented in Matlab R2019b, m language. Parameter setting: the battlefield area is $2000m \times 1800m$. Reconnaissance UAV: the flight speed is $20m/s$, the reconnaissance radius is $500m$, and the maximum flight range is 35km. Attack UAV: the flight speed is $15m/s$ and the maximum flight range is 20km. The first group of experiments is to test the effectiveness and feasibility of the algorithm. In addition, the UAV swarm composed of 2 reconnaissance UAVs and 6 attack UAVs is used to perform reconnaissance attack tasks on 20 ground targets. The second group of experiment is the stability and robustness test of the algorithm. Based on Monte Carlo method, the impact of the change of UAV and target number on the performance of the algorithm is analyzed, and the performance of the algorithm is verified when the UAV swarm scale and the number of targets are large.

5.1. Effectiveness and Feasibility Test. In order to test the effectiveness and feasibility of the proposed method, a UAV swarm composed of two reconnaissance UAVs and six attack UAVs has been used to perform reconnaissance attack tasks on 20 ground targets. At the initial time, the specific parameter settings of UAVs and targets in the swarm are shown in Tables 2 and 3.

In the heterogeneous UAV swarm, two S-UAVs search targets in the 2000×1800 range mission environment according to the multimachine partition and zoning mechanism, and the results are shown in Figure 6. It can be seen intuitively from Figure 6 that, within 8 time periods, 20 targets are detected by S_1 and S_2, respectively, as time goes by.

The BWPA is used to solve the above swarm dynamic task allocation problem. Set the number of wolves $N = 100$, the maximum number of iterations $k_{\max} = 100$, the maximum number of walks $T_{\max} = 100$, the decision distance $d_{\mathrm{near}} = 2$, the walking step $\mathrm{step}_a = 1$, the running step $\mathrm{step}_b = 2$, the attack step $\mathrm{step}_c = 1$, and the update scale factor is 5. As shown in Table 4, we can directly obtain the target number vector $Y = (1, 2, 3, \ldots, 20)$; then the position X_i can be expressed as code $(x_{i1}, x_{i2}, \ldots, x_{ij}, \ldots, x_{im})$, $i = 1, 2, \ldots, N$, $j = 1, 2, \ldots, m$, $N = 100$, $m = 120$, and $x_{ij} \in \{0, 1\}$. By solving the above problems through BWPA, the variation curve of UAV swarm travel (objective function value) with time is displayed in Figure 7.

Table 2: Initial information of UAV swarm.

ID_U	Loc_U	Ru	V (m/s)
S_1	(0, 0)	0	20
S_2	(1000, 0)	0	20
A_1	(0 , 0)	20	15
A_2	(0 , 0)	18	15
A_3	(0 , 0)	16	15
A_4	(0 , 0)	15	15

Table 3: Target initial information.

ID_T	Loc_T	ω_T	RT	ID_T	Loc_T	ω_T	R_T
1	(180 , 800)	7	5	11	(880 , 720)	9	3
2	(220 , 380)	6	2	12	(1060 , 1800)	1	5
3	(240 , 1700)	7	3	13	(1220 , 1400)	6	3
4	(340 , 1240)	8	5	14	(1360 , 360)	4	3
5	(380 , 1240)	9	3	15	(1620 , 400)	5	5
6	(520 , 440)	11	2	16	(1700 , 20)	6	2
7	(600 , 1120)	2	1	17	(1760 , 760)	8	3
8	(740 , 1180)	6	5	18	(1860 , 1100)	3	1
9	(780 , 880)	12	3	19	(1920 , 100)	9	3
10	(820 , 1600)	8	2	20	(1920 , 1780)	7	5

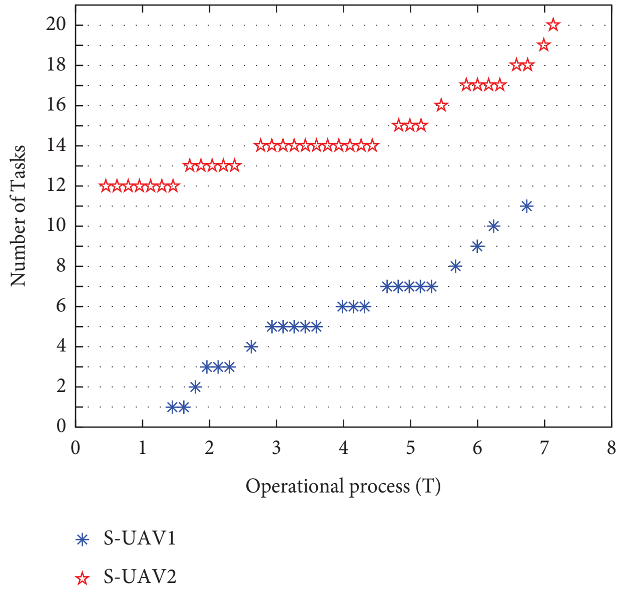


Figure 6: Search results.

Table 4: S-UAV and A-UAV quantity ratio information.

UAV swarm scale	6	7	8	9
Number of S-UAV	2	2	2	2
Number of A-UAV	4	5	6	7

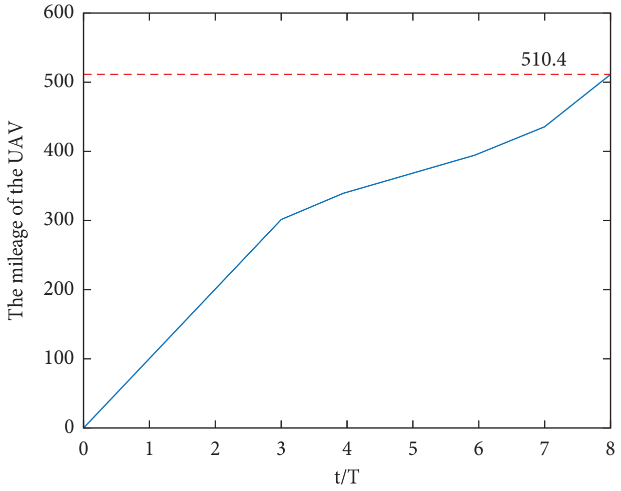


Figure 7: UAV swarm total stroke change curve.

As shown in Figure 7, with the increase of task execution time, the total travel of UAV swarm continues to increase in a broken line. After 8 task cycles, the task is completed, so that the total travel of UAV swarm is 510.4 km. It is consistent with the reality of swarm task execution, indicating that the proposed method is feasible to solve the dynamic task allocation problem.

Figures 6 and 8 visually show the whole process of task allocation, coordination, cooperation, and execution of heterogeneous UAV swarm starting from the base after 8 time cycles. From the figure, you can master the position, moving track, assigned task, task execution status, and other information of UAV in the task environment within 2000×1800 at any time. When $t = 0 \sim T$, Task_{12} is detected by S_2. According to the execution cost-effectiveness ratio of each UAV, A_1 executes it, and the other UAVs turn into standby state. When $t = T \sim 2T$, Task_1, Task_2, and Task_3 are detected by S_1 and Task_{13} is detected by S_2. Task_1, Task_2, Task_3, and Task_{13} are allocated to A_4, A_1, A_3, and A_2, respectively, according to the cost-effectiveness ratio of each A-UAV. When $t = 2T \sim 3T$, Task_4 and Task_5 are detected by S_1 in turn, and Task_{14} is detected by S_2. Since the cost-effectiveness ratio of A_1, A_2, and A_3 is better than that of A_4, Task_4, Task_5, and Task_{14} are allocated to A_2, A_1, and A_3, respectively, and A_4 turns into standby state. When $t = 3T \sim 4T$, Task_6 is found by S_1, and the efficiency cost ratio of Task_6 executed by A_1 is significantly better than that of other A-UAVs. Therefore, A_1 executes Task_6, and the other A-UAVs are in a standby state. When $t = 4T \sim 5T$, Task_5 and Task_7 are found by S_1. According to the allocation mechanism based on the implementation efficiency cost ratio, Task_5 and Task_7 are allocated by A_2 and A_1. When $t = 5T \sim 6T$, Task_9 is detected by S_1, and Task_{17} and Task_{16} are detected by S_2 and then executed by A_1, A_3, and A_2 in the swarm, respectively. When $t = 6T \sim 7T$, Task_{11} is detected by S_1, and Task_{18} and Task_{19} are detected by S_2 in turn and then executed by A_3, A_2, and A_1 in the swarm, respectively. When $t = 7T \sim 8T$, Task_8 and Task_{10} are detected by S_1, and Task_{20} is detected by S_2 and then executed by UAV_3, UAV_1, and UAV_2 in the swarm, respectively. In the whole task execution process, the characteristics of UAV are fully utilized. The task allocation is coherent and the task execution process is compact. There is no spare time waiting for UAV during the execution of each task.

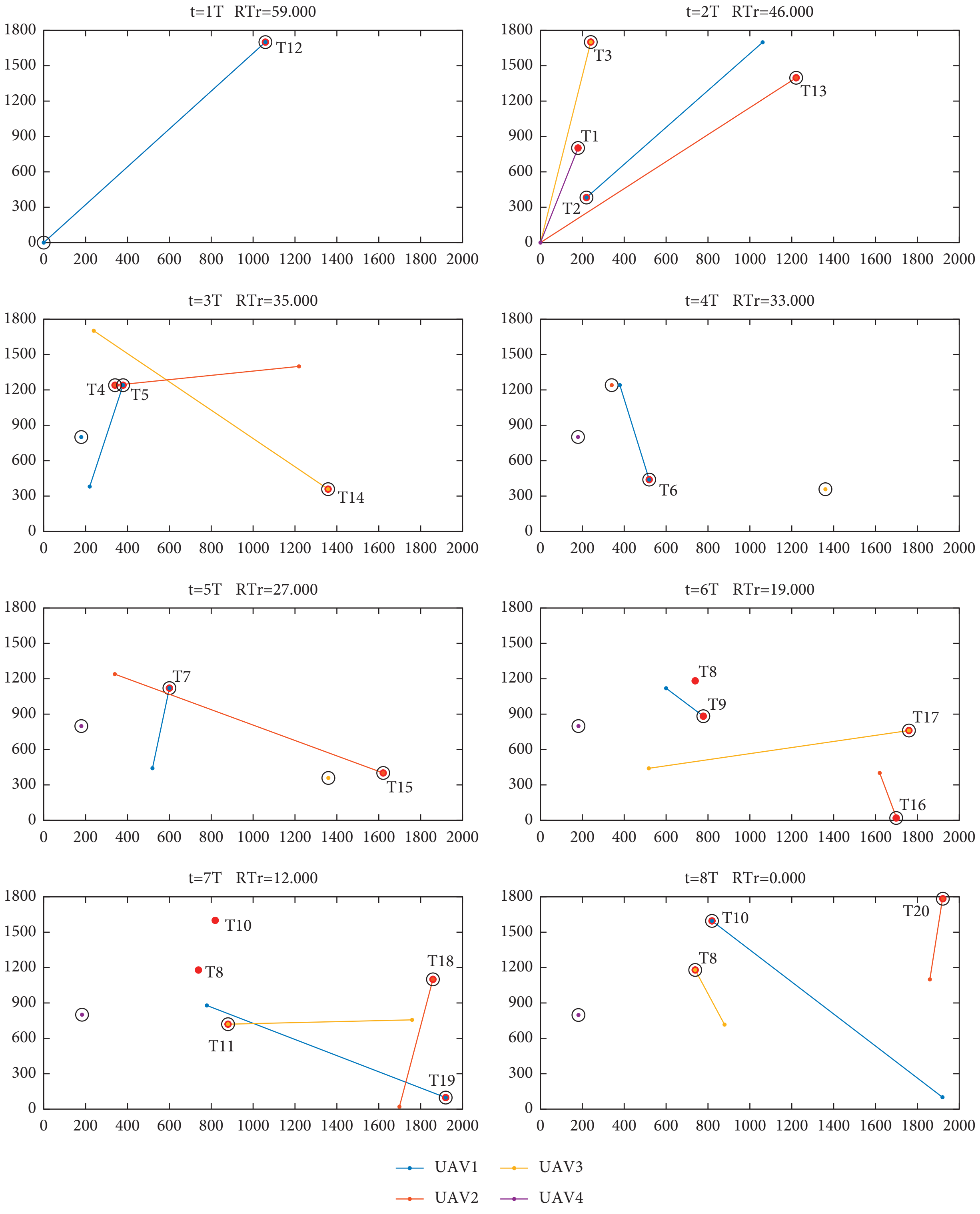


Figure 8: Attack UAV mission progress.

Figure 9 shows the resource changes of each UAV within 8 units of time during which the UAV swarm performs tasks. It is obvious that the task load distribution of UAV is reasonable, the resource consumption is relatively balanced, and the work does not interfere with each other. That is to say, the proposed method improves the resource utilization rate of UAV, gives full play to the advantages of UAV collaborative work, realizes the complementarity of capabilities and resources, and helps maintain the long-term execution ability to deal with emergencies.

To sum up, when the total resources carried by the UAV meet the needs of the current task, in the face of the searched

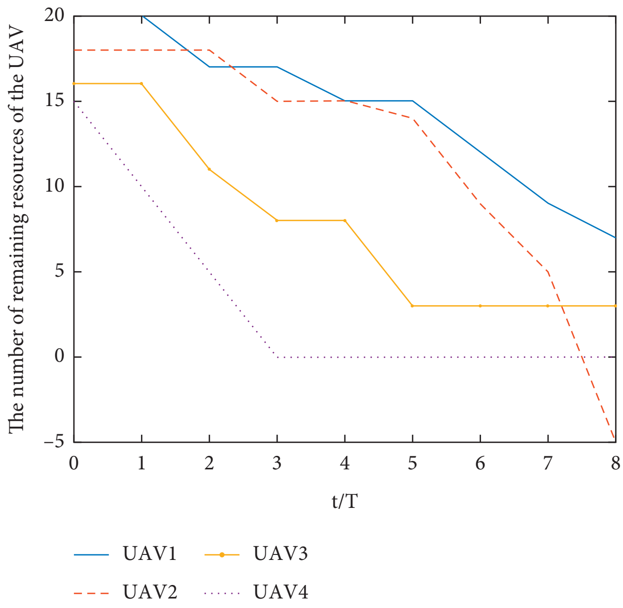


Figure 9: UAV resource change curve.

target, the UAV swarm can comprehensively analyze the factors such as the number of resources owned by the UAV, task distance, task value, and resources required by the task according to the current situation, timely efficiently allocate the task to each UAV, execute it quickly, and realize full dynamic task allocation. Apart from that, it fully reflects the high coordination of multiple UAVs, ensures the maximum net benefit of executing tasks (i.e., maximizing the benefit of executing tasks and minimizing the cost of executing tasks), and reflects the rationality of this allocation method.

Table 5: Initial information of UAV swarm.

ID_U	Loc_U	Ru	V (m/s)
S_1	(0, 0)	0	20
S_2	(1000, 0)	0	20
A_1	(0 , 0)	20	15
A_2	(0 , 0)	18	15
A_3	(0 , 0)	16	15
A_4	(0 , 0)	15	15
A_5	(500 , 0)	20	15
A_6	(1000 , 0)	18	15
A_7	(2000 , 0)	16	15

5.2. Stability and Robustness Test. In order to further test the stability and robustness of the proposed algorithm, Monte Carlo experimental method is used to set the number of tasks as 10, 15, and 20, respectively, and the number of UAVs in the cluster as 6, 10, 12, and 15, respectively. The ratio of S-UAV and A-UAV in UAV swarm is shown in Table 4. Beyond that, the relevant parameters of the task are the same as those in the previous section. The tasks of the four groups are $\text{Task}_1 \sim \text{Task}_{10}$, $\text{Task}_1 \sim \text{Task}_{15}$, and $\text{Task}_1 \sim \text{Task}_{20}$, respectively. The swarm parameters are displayed in Table 5. The swarm members of the four groups are $S_1 \sim S_2$ and $A_1 \sim A_4$, $S_1 \sim S_3$ and $A_1 \sim A_6$, $S_1 \sim S_4$ and $A_1 \sim A_8$, and $S_1 \sim S_5$ and $A_1 \sim A_{10}$, respectively.

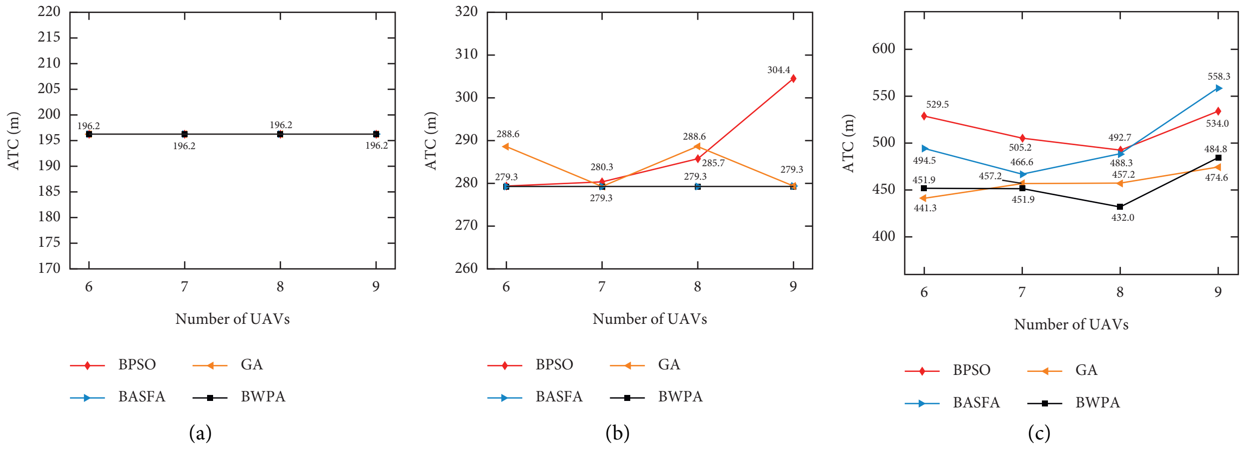
The method proposed in this paper, together with binary particle swarm optimization (BPSO) [37], binary artificial fish swarm algorithm (BASFA) [38, 39], and genetic algorithm (GA) [5, 40], respectively, runs the above grouping experiments 50 times independently, and the average track cost (ATC) and the average total calculation time (ATCT) are counted. ATC is the average value of the UAV trajectory where all targets have been destroyed. ATCT is the average of the time taken for all targets to be destroyed. Simulation termination condition: all tasks are completed (i.e., all task resource requirements are 0). The four solution methods choose the same general parameters. For example: the population size is $N = 50$ (wolf swarm, particle swarm, fish swarm, etc.); the maximum number of iterations is $k_{\max} = 100$. The other parameters are set according to [37], [38], and [40], as shown in Table 6.

Through simulation experiments, the ACT results of the four methods are displayed in Figures 10(a) ~10(c).

As shown in Figure 10, when the target number is 5, no matter how the swarm size changes, the four methods achieve the same results. In other words, when the number of targets is small, the size of the swarm has less impact on the four methods to solve ACT, and the established model has good adaptability to the algorithm. When the target number is 10, the change in swarm size has a particularly significant impact on the ACT required by BPSO and GA, but the proposed method and BASFA are still unaffected. When the target number is 15, the four methods are affected to varying degrees, but the size fluctuation range of the ACT of the proposed method is small, which is significantly better than that of other methods. The results show that the proposed method has good robustness and strong adaptability to the environment. In general, the number of UAVs has no significant impact on the ACT of the proposed

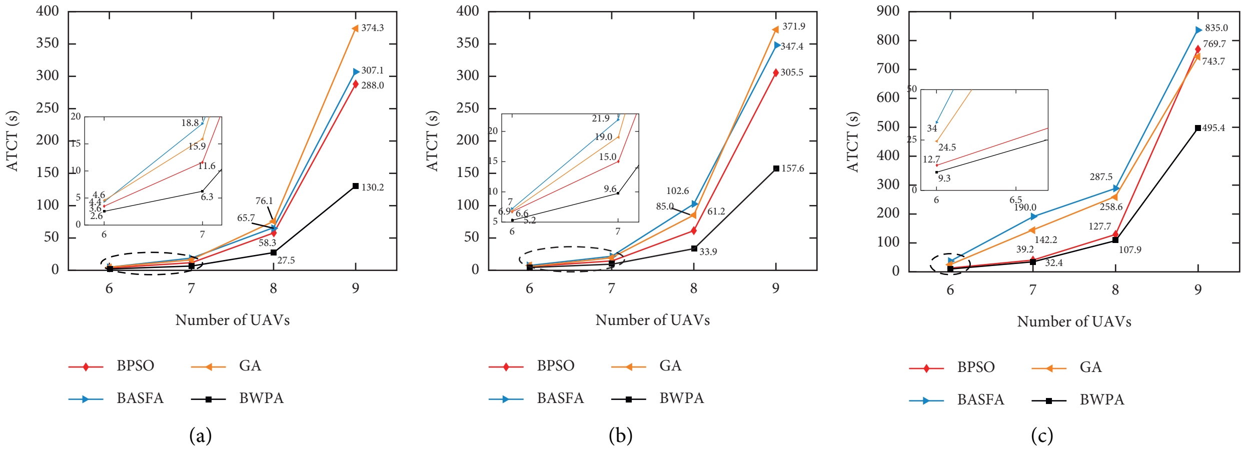
Table 6: Solving method parameter settings.

Method	Parameter settings
BWPA	Maximum number of walks: $T_{\max} = 10$, walking step length: $\text{step}_a = 1$, strike step length： $\text{step}_b = 2$, attack step length： $\text{step}_c = 1$, decision distance： $d_{\text{near}} = 2$, update scale factor： $\beta = 5$.
BPSO	Inertia weight： $\omega = 0.7298$, learning factor ： $c_1 = c_2 = 1.4946$, individual speed limit ：(-0.5,0.5).
BASFA	Maximum number of trials: $\text{try}_{\text{num}} = 100$, perception distance： $v = 1$, congestion factor： $de = 0.618$, walking step length： $\text{step} = 1$.
GA	Crossover probability: $P_c = 0.9$, mutation probability: $P_m = 0.03$.



(a) (b) (c)

Figure 10: The impact of the UAVs number and the targets number on four algorithms' ACT result.



(a) (b) (c)

Figure 11: The impact of UAVs number and the target number on four algorithms' ATCT result.

algorithm. The proposed algorithm can achieve a relatively ideal ACT in various situations and show good robustness. The results of the proposed model by other algorithms fully reflect the good adaptability of the designed model.

As shown in Figure 11, when the number of UAVs is constant, the ATCT of the four algorithms increases as the number of targets increases. When the number of targets is constant, the ATCT increases with the change in the number of UAVs. The change trend of ATCT corresponding to the four methods in Figures 11(a) ~ 11(c) is basically the same. When the number of UAVs is 9, the ATCT increment of the four methods is the largest, but the ATCT of BWPA is always better than that of other methods. It shows that the proposed method has high solution efficiency and better dynamic adaptability than other methods. When the target number is the same as the swarm size, the ATCT of other methods is about twice or even greater than that of BWPA. In contrast, BWPA shows better robustness. In general, the proposed method can achieve the best ACTC results in various situations and has good real-time performance.

In summary, through comparative experiments, it can be seen that the proposed model can use most algorithms to obtain satisfactory results, showing the applicability of the proposed model. As far as ACT and ATCT are concerned, the proposed algorithm performs better than the comparison algorithm in terms of UAV task load and resource utilization balance, calculation speed, and solution efficiency. The impact of UAV and the number of targets on the algorithm is not significant, indicating that the method has good stability, dynamic adaptability, and scalability and can better meet the needs of dynamic allocation.

6. Conclusion

This paper investigates the dynamic task allocation of UAV swarm. According to the real-time requirements of operation, a dynamic task allocation model of multiobjective, multitask, heterogeneous multiaircraft platform and multiconstraints has been designed based on target cost-effectiveness ratio and task execution time window. Apart from that, the designed model adopts the mode of attacking while searching and giving priority to important targets. The information acquisition of battlefield targets completely depends on the reconnaissance UAV to search targets based on the search strategy and generates the task sequence to be attacked in each task cycle in line with the cost-effectiveness ratio and task execution time window. Furthermore, the task allocation scheme adopts one-dimensional 0–1 coding, and then the dynamic allocation model is solved on account of BWPA. Finally, two groups of simulation experiments are used to verify the effectiveness and dynamic adaptability of the proposed method. According to the results, the designed task allocation method not only has the good adaptability to the change of target and UAV number, as well as good stability and scalability, but also can effectively solve the dynamic task allocation problem of heterogeneous UAV swarm in unknown environment. Therefore, the established model and solution method can provide a useful reference for task allocation and other related problems. However, this paper only considers the situation when the target is static but fails to consider the constraints such as the threat of the target, which results in some limitations. Thus, the dynamic task allocation of UAV swarm with constraints such as dynamic targets and target threats will be studied further.

Data Availability

The data used to support the findings of this study are included in the article and are available from the corresponding authors upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the Military Science Project of National Social Science Foundation (2019-SKJJ-C-092), Natural Science Foundation of Shaan Xi Province (no. 2020JQ-493), Military Equipment Research Project (WJ2020A020029), and Equipment Comprehensive Research Project (WJY20211A030018).

References

[1] H. Bui, X. Han, and S. Mandal, “Optimization-based decision support algorithms for a team-in-loop planning experiment,” in *Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 4684–4689, San Antonic, TX, USA, 2009.

[2] Q. Peng, H. Wu, and R. Xue, “Review of dynamic task allocation methods for UAV swarms oriented to ground targets,” *Complex System Modeling and Simulation*, vol. 1, no. 3, pp. 163–175, 2021.

[3] D. Zou, H. Liu, and L. Gao, “An improved differential evolution algorithm for the task assignment problem,” *Engineering Applications of Artificial Intelligence*, vol. 24, no. 4, pp. 616–624, 2011.

[4] G. W. Jia and J. F. Wang, “Review of UAV swarm mission planning methods,” *Systems engineering and electronic technology*, vol. 43, no. 1, pp. 99–111, 2021.

[5] T. Shima, S. J. Rasmussen, A. G. Sparks, and K. M. Passino, “Multiple task assignments for cooperating uninhabited aerial vehicles using genetic algorithms,” *Computers & Operations Research*, vol. 33, no. 11, pp. 3252–3269, 2006.

[6] F. Su, Y. Chen, and L. C. Shen, “UAV cooperative multi task assignment based on ant colony algorithm,” *Acta Aeronautica Sinica*, vol. 29, no. 1, pp. 184–191, 2008.

[7] L. Shi, C. S. Liu, and Q. Ye, “Dynamic resource allocation of UAV cooperative attack based on PSO,” *Flight mechanics*, vol. 32, no. 4, pp. 368–371, 2014.

[8] X. L. Zhao, K. W. Zhang, and Z. Z. Li, “Research on dynamic reconnaissance resource allocation of multiple UAVs,” *Electronics Optics and Control*, vol. 27, no. 6, pp. 11–15, 2020.

[9] L. Cao, H. S. Tan, and H. Peng, “A hybrid dynamic task allocation method for multiple UAVs,” *Journal of Nanjing University of Technology*, vol. 39, no. 2, pp. 206–214, 2015.

[10] Z. Liu, W. Li, and J. C. Ren, “Modeling and solving method of multi base and multi UCAV task allocation,” *Journal of Southeast University*, vol. 49, no. 1, pp. 91–96, 2019.

[11] Z. Liu, X. G. Gao, X. W. Fu, and Z.-Y. Mou, “Coalition formation of multiple heterogeneous unmanned aerial vehicles in cooperative search and attack in unknown environment,” *Acta Armamentarii*, vol. 36, no. 12, pp. 2284–2297, 2015.

[12] X. Chen and T. Tang, “Dynamic task allocation method for multiple UAVs in uncertain environment,” *Firepower and command control*, vol. 38, no. 1, pp. 45–49, 2013.

[13] F. Yan, X. Q. Zhu, and Z. Zhou, “Real-time task allocation for a heterogeneous multi-UAV simultaneous attack,” *Sci Sin Inform*, vol. 49, no. 5, pp. 555–569, 2019.

[14] M. Y. Zhang, M. Y. Wang, and X. D. Wang, “Cooperative real-time task assignment of UAV group based on improved contract net,” *Aero Weaponry*, vol. 26, no. 4, pp. 38–46, 2019.

[15] Y. Q. Wang, A. Zhang, and B. I. Wen-Hao, “Mission planning of manned/unmanned aerial vehicle formation for Time Critical Target attacking,” *Electronics Optics and Control*, vol. 25, no. 8, pp. 7–10, 2018.

[16] S. C. Xu, Q. Wang, and X. M. Liu, “Multi UAV task redistribution based on split and join particle swarm optimization,” *Firepower and command and control*, vol. 37, no. 4, pp. 188–191, 2012.

[17] J. Ren, X. G. Gao, and J. S. Zheng, “UAV mission decision model in complex environment,” *System engineering and electronic technology*, vol. 32, no. 1, pp. 100–103, 2010.

[18] H. Duan, L. Pei, and Y. Yu, "A predator-prey particle swarm optimization approach to multiple UCAV air combat modeled by dynamic game theory," *IEEE/CAA Journal of Automation*, vol. 2, no. 1, pp. 11–18, 2015.

[19] S. J. Yang, S. W. Wang, and J. Tao, "Research on multi UCAV cooperative task allocation in dynamic environment," *Electronics Optics and Control*, vol. 19, no. 7, pp. 32–36, 2012.

[20] H. S. Wu, F. G. Zhang, R. J. Zhan, and S. Wang, "A binary wolf pack algorithm for solving 0-1 knapsack problem," *Systems Engineering and Electronics*, vol. 36, no. 8, pp. 1660–1667, 2014.

[21] C. Li, K. Song, and L. Yang, "Low computational complexity design over sparse channel estimator in underwater acoustic OFDM communication system," *IET Communications*, vol. 11, no. 7, pp. 1143–1151, 2017.

[22] H. Dai, Y. Huang, C. Li, and Y. Luxi, "Energy-efficient resource allocation for device-to-device communication with WPT," *IET Communications*, vol. 11, no. 3, pp. 326–334, 2017.

[23] Y. Wu, X. Guan, W. Yang, and Q. Wu, "UAV swarm communication under malicious jamming: joint trajectory and clustering design," *IEEE Wireless Communications Letters*, vol. 10, no. 10, pp. 2264–2268, 2021.

[24] Z. Fang, J. Wang, Y. Ren, Z. Han, H. V. Poor, and L. Hanzo, "Age of information in energy harvesting aided massive multiple access networks," *IEEE Journal on Selected Areas in Communications*, 2022.

[25] J. Wang, C. Jiang, Z. Wei, C. Pan, H. Zhang, and Y. Ren, "Joint UAV hovering altitude and power control for space-air-ground IoT networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1741–1753, 2018.

[26] J. Tian, *Research on Modeling and Optimization Technology of Multi-UAV Cooperative Reconnaissance Mission Planning*, National University of Defense Technology, Changsha, China, 2007.

[27] X. Q. Li, X. X. Sun, and D. Wang, "UCAV dynamic task allocation model and research based on genetic algorithm," *Journal of System Simulation*, vol. 16, pp. 4387–4389, 2008.

[28] H. S. Wu, F. M. Zhang, and L. S. Wu, "A new swarm intelligence algorithm-wolf pack algorithm," *Systems Engineering and Electronics Technology*, vol. 35, no. 11, pp. 2430–2438, 2013.

[29] T. H. Yi, C. W. Wang, and H. N. Li, "Hierarchic wolf algorithm for optimal triaxial sensor placement," *Journal of Building Structures*, vol. 35, no. 4, pp. 223–229, 2014.

[30] H. S. Wu, F. M. Zhang, H. Li, and X.-L. Liang, "Discrete wolf pack algorithm for traveling salesman problem," *Control and Decision*, vol. 30, no. 10, pp. 1861–1867, 2015.

[31] Y. Q. Yan, *Research on AGV Path Planning Algorithm Based on Swarm Intelligence Optimization*, Jilin University, Changchun, China, 2017.

[32] Y. Ye and H. Z. Zhang, "Wolf group algorithm for multi-distribution center vehicle routing problem," *Journal of Computer Applications*, vol. 34, no. 9, pp. 2590–2593, 2017.

[33] Y. L. Liu, W. M. Li, H. S. Wu, and W. Song, "Track planning for unmanned aerial vehicles based on wolf pack algorithm," *Journal of System Simulation*, vol. 27, no. 8, pp. 1838–1844, 2015.

[34] J. Q. Wang, Y. Y. Jia, and Q. Y. Xiao, "Application of wolf pack search algorithm to optimal operation of hydropower station," *Advances in Science and Technology of Water Resources*, vol. 35, no. 3, pp. 1–4, 2015.

[35] R. X. Wei and Z. C. Wu, "Study on task allocation of UAV swarm based on cognitive control," *Journal of System Simulation*, vol. 33, no. 7, pp. 1574–1581, 2021.

[36] X. L. Li, "An optimizing method based on autonomous animats: fish-swarm algorithm," *Systems Engineering-Theory & Prac-tice*, vol. 22, no. 11, pp. 32–38, 2002.

[37] Y. W. Zhang, L. Wang, and Q. D. Wu, "Dynamic adaptation cuckoo search algorithm," *Control and Decision*, vol. 29, no. 4, pp. 617–622, 2014.

[38] H. Zhou and L. I. Yu, "Cuckoo search algorithm with dynamic inertia weight," *CAAI Transactions on Intelligent Systems*, vol. 10, no. 4, pp. 645–651, 2015.

[39] M. A. K. Azad, A. M. A. Rocha, and E. M. Fernandes, "A simplified binary artificial fish swarm algorithm for 0–1 quadratic knapsack problems," *Journal of Computational and Applied Mathematics*, vol. 259, pp. 897–904, 2014.

[40] H. D. Mathias, A. S. Wu, and D. Dang, "Analysis of evolved response thresholds for decentralized dynamic task allocation," 2021, https://arxiv.org/abs/2106.06019.